

TUGAS AKHIR - KS 141501

**PENGOPTIMALAN BIAYA TRANSPORTASI
MENGUNAKAN *METODE LEAST COST* DAN
METODE STEPPING STONE DENGAN BAHASA
PEMROGRAMAN JAVA UNTUK DISTRIBUSI BBM
STUDI KASUS: PT. PERTAMINA PATRA NIAGA**

***OPTIMIZING TRANSPORTATION COSTS USING
LEAST COST METHOD AND STEPPING STONE
METHOD WITH JAVA PROGRAMMING
LANGUAGE FOR FUELS DISTRIBUTION
CASE STUDY: PT. PERTAMINA PATRA NIAGA***

Syafriandi Nurrahman
NRP 05211140000136

Dosen Pembimbing
Ahmad Mukhlason, S.Kom., M.Sc.
DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KS 141501

**PENGOPTIMALAN BIAYA TRANSPORTASI
MENGUNAKAN METODE LEAST COST DAN
METODE STEPPING STONE DENGAN BAHASA
PEMROGRAMAN JAVA UNTUK DISTRIBUSI BBM
STUDI KASUS: PT. PERTAMINA PATRA NIAGA**

Syafriandi Nurrahman
NRP 05211140000136

Dosen Pembimbing
Ahmad Mukhlason, S.Kom., M.Sc.
DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



ITS

Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - KS 1501

**OPTIMIZING TRANSPORTATION COSTS USING
LEAST COST METHOD AND STEPPING STONE
METHOD WITH JAVA PROGRAMMING
LANGUAGE FOR FUELS DISTRIBUTION
CASE STUDY: PT. PERTAMINA PATRA NIAGA**

Syafriandi Nurrahman
NRP 05211140000136

Supervisor

Ahmad Mukhlason, S.Kom., M.Sc.

DEPARTMENT OF INFORMATION SYSTEMS

Faculty of Information Technology and Communication

Institut Teknologi Sepuluh Nopember

Surabaya 2018

LEMBAR PENGESAHAN

**PENGOPTIMALAN BIAYA TRANSPORTASI
MENGUNAKAN METODE LEAST COST DAN
METODE STEPPING STONE DENGAN BAHASA
PEMROGRAMAN JAVA UNTUK DISTRIBUSI BBM
STUDI KASUS: PT. PERTAMINA PATRA NIAGA**

Disusun untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Departemen Sistem Informasi
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh :

Syafriandi Nurrahman
NRP 05211140000136

Surabaya, 20 Juli 2018

**KEPALA
DEPARTEMEN SISTEM INFORMASI**



Dr. Ir. Aris Tjahyanto, M.Kom.
NIP 19730219 199802 1 001

**PENGOPTIMALAN BIAYA TRANSPORTASI
MENGUNAKAN METODE LEAST COST DAN
METODE STEPPING STONE DENGAN BAHASA
PEMROGRAMAN JAVA UNTUK DISTRIBUSI BBM
STUDI KASUS: PT. PERTAMINA PATRA NIAGA
TUGAS AKHIR**

(Penguji II)

**PENGOPTIMALAN BIAYA TRANSPORTASI
MENGUNAKAN METODE LEAST COST DAN
METODE STEPPING STONE DENGAN BAHASA
PEMROGRAMAN JAVA UNTUK DISTRIBUSI BBM
STUDI KASUS: PT. PERTAMINA PATRA NIAGA**

Nama Mahasiswa : Syafriandi Nurrahman
NRP : 05211140000136
Departemen : Sistem Informasi FTIF-ITS
Pembimbing 1 : Ahmad Mukhlason, S.Kom., M.Sc.

ABSTRAK

PT. Pertamina Patra Niaga sudah melakukan Fuel Trading dari tahun 2004 dan kemudian mendistribusikannya ke seluruh Indonesia. Sementara biaya yang digunakan untuk distribusi pasti tidaklah sedikit. Disini penulis ingin mengoptimalkan biaya distribusi dengan metode yang ada dalam Riset Operasi.

Sementara dalam Riset Operasi masalah transportasi dibahas dalam Linear Programming. Metode yang dipilih adalah Least Cost (LC) dan Stepping Stone (SS), dimana metode LC dilakukan terlebih dahulu setelah itu baru dilakukan dengan metode Stepping Stone. Karena pada dasarnya prinsip metode LC yang mengalokasikan supply dari biaya pengiriman terendah matriks transportasi untuk memenuhi demand belum optimal, maka dilakukan metode berikutnya yaitu metode SS. Bunker ditetapkan menjadi origin dari distribusi dan kapasitas bunker disebut supply, serta kota/kabupaten ditetapkan menjadi destination dan permintaanya dinamakan demand.

Sesuai dengan langkah diatas, kedua metode tersebut dapat dilakukan untuk menyelesaikan masalah distribusi tersebut. Dan dengan membandingkan biaya eksisting dengan biaya setelah diterapkan metode tersebut, dapat diketahui bahwa kedua metode

tersebut tidak hanya dapat meminimalkan biaya distribusi, tetapi juga dapat mengalokasikan BBM dengan tepat sesuai dengan demand yang dibutuhkan. Serta dengan adanya User Interface yang telah dibuat dengan bahasa pemrograman Java, dapat membantu mempermudah perhitungan dengan kedua metode tersebut.

Kata Kunci : Riset Operasi, Linear Programming, *Least Cost*, *Stepping Stone*, Optimasi, Java

**OPTIMIZING TRANSPORTATION COSTS USING LEAST
COST METHOD AND STEPPING STONE METHOD
WITH JAVA PROGRAMMING LANGUAGE FOR FUELS
DISTRIBUTION
CASE STUDY: PT. PERTAMINA PATRA NIAGA**

Student Name : Syafriandi Nurrahman
NRP : 05211140000136
Department : Sistem Informasi FTIF-ITS
Supervisor 1 : Ahmad Mukhlason, S.Kom., M.Sc.

ABSTRACT

PT. Pertamina Patra Niaga has been doing Fuel Trading since 2004 and distributed it to all over Indonesia. While the cost used for distribution is definitely not small. Here the authors want to optimize the cost of distribution with existing methods in Operations Research.

While in Operations Research the problem of transportation is discussed in Linear Programming. The selected method is Least Cost (LC) and Stepping Stone (SS), where the LC method is done first after that proceed to Stepping Stone method. Because basically the principle of LC method which allocates supply from the lowest delivery cost of transportation matrix to fulfill the demand is not optimal yet, furthermore the SS method is required to achieve the optimal solution. The bunkers are set to be the origin of the distribution and the bunker capacity is called supply, and the city / district is set to be the destination and the demand is called demand itself.

In accordance with the steps above, both methods can be done to solve the distribution problem. And by comparing the existing costs

with costs after the method is applied, it can be seen that the two methods can not only minimize the cost of distribution, but also can allocate fuel exactly in accordance with demand required. And with the User Interface that has been created with the Java programming language, can help simplify the calculation with both methods.

Keywords : Operation Research, Linier Programming, Least Cost, Stepping Stone, Optimization, Java

KATA PENGANTAR

Puji syukur kehadirat Allah SWT atas limpahan rahmat dan karunia-Nya sehingga Penulis mendapatkan kelancaran dalam menyelesaikan laporan Tugas Akhir dengan judul:

**PENGOPTIMALAN BIAYA TRANSPORTASI
MENGUNAKAN METODE LEAST COST DAN METODE
STEPPING STONE DENGAN BAHASA PEMROGRAMAN
JAVA UNTUK DISTRIBUSI BBM STUDI KASUS: PT.
PERTAMINA PATRA NIAGA**

yang disusun dalam rangka memenuhi salah satu syarat untuk memperoleh gelar sarjana pada Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya.

Dalam pelaksanaan dan pembuatan Tugas Akhir ini tentunya sangat banyak bantuan yang penulis terima dari berbagai pihak, oleh karena itu penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada:

- 1) Allah SWT yang selalu memberikan Rahmat dan Karunia-Nya, sehingga penulis dapat dengan lancar menyelesaikan tugas akhir ini.
- 2) Orang tua penulis, Moch. Syaiful Anwar dan Wilujeng Harimulyati Agustini yang selalu memberi dukungan moril dan material serta selalu memanjatkan doa yang tiada hentinya hingga penulis dapat menyelesaikan pendidikan sejauh ini.
- 3) Bapak Ahmad Mukhlason, S.Kom., M.Sc. selaku dosen pembimbing pertama dan satu-satunya yang telah dengan sabar membimbing penulis untuk menyelesaikan Tugas Akhir.
- 4) Bapak Edwin Riksakomara dan Ibu Wiwik Anggraeni, S.Si., M.Kom. selaku dosen penguji yang telah memberikan masukan dan saran kepada Tugas Akhir penulis.

- 5) Mas Zulfikar, Mas Bombom, dan Mbak Lia selaku pembimbing lapangan dari PT. Pertamina Patra Niaga yang sangat membantu penulis untuk menyelesaikan Tugas Akhir ini.
- 6) Bapak Nisfu Asrul Sani S.Kom., M.Sc. selaku dosen wali penulis yang telah memberikan banyak bantuan berupa saran, masukan dan bimbingan tambahan serta jasa-jasa lainnya yang tidak bisa disebutkan semua selama penulis berkuliah di Departemen Sistem Informasi ITS.
- 7) Kepala Departemen, Bapak Ir. Aris Tjahyanto, M.Kom.
- 8) Seluruh dosen pengajar beserta staf dan karyawan di Departemen Sistem Informasi, Fakultas Teknologi Informasi dan komunikasi ITS Surabaya.
- 9) Kakak saya Zulfikar Al-Azhari dan Zulfa Tisa Shafira yang terus mengingatkan dan membantu diluar dan didalam perkuliahan
- 10) Laras Binarwati S.Si, selaku sahabat sejati saya yang terus memberikan semangat untuk menyelesaikan kuliah
- 11) Mas Dewangga dewa koding yang memberi saya pencerahan dalam pengerjaan
- 12) Sahabat-sahabat penulis yang tidak memberikan support apa-apa seperti Dimas, Ambon Fariz, Bima Beam, Shadiq Adiq, Dani Coco, Bagus Parto.
- 13) Dan teman-teman remajaId yang terus menghibur
- 14) Juga kepada semua pihak yang tidak sempat disebutkan satu per satu yang telah membantu penyelesaian Tugas Akhir ini.

Terima kasih atas segala bantuan, dukungan, serta doa yang telah diberikan. Penulis sangat sadar bahwa tugas akhir ini masih memiliki kekurangan dan ketidaksempurnaan dalam penulisan, dengan begitu penulis mengharapkan kritik dan saran dari pembaca yang bersifat membangun sebagai bahan acuan penelitian-

penelitian selanjutnya. Semoga tugas akhir ini dapat memberikan manfaat yang besar dan ilmu pengetahuan untuk menambah wawasan bagi para pembaca.

Surabaya, Juli 2018

Syafriandi Nurrahman

Halaman ini sengaja dikosongkan

DAFTAR ISI

ABSTRAK	i
ABSTRACT	iii
KATA PENGANTAR	v
DAFTAR ISI	ix
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xvii
DAFTAR KODE PROGRAM	xviii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
1.6 Relevansi	4
BAB II TINJAUAN PUSTAKA	5
2. 1 Penelitian Sebelumnya	5
2. 2 Dasar Teori	6
2.2.1 Vehicle Routing Problem	6
2.2.2 Riset Operasi	8
2.2.3 Linear Programming	11
2.2.4 Metode Transportasi	11
2.2.5 Formulasi Model Matematis	13
2.2.6 <i>Least Cost</i>	15
2.2.7 <i>Stepping Stone</i>	15
2.2.8 Java Programming	16

BAB III	METODOLOGI PENELITIAN	17
3.1	Studi Literatur	18
3.2	Pengumpulan Data	18
3.3	<i>Interview</i>	19
3.4	Pra-proses Data	19
3.5	Penyusunan Model Matematis	20
3.6	Pengimplementasian Java pada Studi Kasus.....	20
3.7	Pembentukan Matriks Transportasi pada Java	20
3.8	Analisis Hasil	20
3.9	Penarikan Kesimpulan dan Saran.....	21
BAB IV	PERANCANGAN.....	23
4.1	Pra-Proses data	23
4.2	Data yang digunakan.....	25
4.2.1	Data <i>Supplier</i>	26
4.2.2	Data <i>Demand</i>	26
4.2.3	Data penghitungan beban biaya distribusi	28
4.3	Alur Distribusi.....	31
4.4	Penyusunan Model Matematis	32
BAB V	IMPLEMENTASI	35
5.1	Lingkungan Uji Coba.....	35
5.2	Pengimplementasian Java	36
5.3	Pembentukan Matriks Awal	36
5.3.1	Algoritma Matriks	36
5.3.2	Metode Input	37
5.4	Algoritma <i>Least Cost</i>	37
5.5	Algoritma <i>Stepping Stone</i>	39
5.5.1	Fix Degenerate Case	41

5.5.2	Penentuan Sel Matriks <i>Stepping Stone</i>	41
5.6	Pengalokasian Demand pada Matriks	42
5.7	Algoritma Solusi Tujuan.....	43
5.8	Flow Pengolahan Data	44
5.8.1	Flow Metode Input	45
5.8.2	Flow Algoritma Matriks	46
5.8.3	Flow Algoritma <i>Least Cost</i>	46
5.8.4	Flow Algoritma <i>Stepping Stone</i>	46
5.8.5	Flow Penentuan Sel <i>Stepping Stone</i>	46
5.8.6	Flow Fix Degenerate Case	47
5.8.7	Flow Algoritma Solusi Tujuan	47
5.9	User Interface (UI) Java.....	47
5.10	Fungsi-Fungsi pada UI <i>Least Cost & Stepping Stone</i> 48	
5.10.1	Fungsi Pembentukan Matriks dan File Masukkan..	48
5.10.2	Fungsi Download Contoh Format.....	50
5.10.3	Fungsi Matriks <i>Least Cost</i>	51
5.10.4	Fungsi Matriks <i>Stepping Stone</i>	51
5.10.5	Fungsi Reset Table	52
5.11	Validasi Java dan Unit Testing	52
5.11.1	Perhitungan <i>Least Cost</i>	52
5.11.2	Perhitungan <i>Stepping Stone</i>	54
5.11.3	Java <i>Stepping Stone Loop Output</i>	55
BAB VI	HASIL DAN PEMBAHASAN	59
6.1	Masukkan Metode <i>Least Cost & Stepping Stone</i>	59
6.2	Pembentukan Matriks <i>Least Cost</i>	61
6.3	Pembentukan Matriks <i>Stepping Stone</i>	64

6.4	Hasil Alokasi Metode <i>Least Cost</i>	67
6.5	Hasil Alokasi Metode <i>Stepping Stone</i>	69
6.6	Perbandingan Biaya Eksisting dengan Biaya Hasil Metode Optimasi Transportasi	70
BAB VII KESIMPULAN DAN SARAN.....		73
7.1	Kesimpulan	73
7.2	Saran.....	73
DAFTAR PUSTAKA.....		75
BIODATA PENULIS.....		77
LAMPIRAN A		79
LAMPIRAN B.....		85

DAFTAR GAMBAR

Gambar 2.1 Permodelan OR	10
Gambar 2.2 Distribusi Model Transportasi.....	12
Gambar 2.3 Matriks Transportasi.....	13
Gambar 3.1 Diagram Alur Pengerjaan Tugas Akhir.....	17
Gambar 3.2 Contoh Data Realisasi HSD.....	19
Gambar 4.1 Contoh x dan y Pivot Tabel.....	24
Gambar 4.2 Contoh Hasil Pivot Tabel.....	25
Gambar 4.3 Perusahaan yang Melakukan Permintaan Pembelian/Demand.....	27
Gambar 4.4 Contoh <i>Destination</i> dan Demand Perusahaan.....	28
Gambar 5.1 Flow Olah Data	45
Gambar 5.2 UI <i>Least Cost & Stepping Stone</i>	48
Gambar 5.3 UI Choose File dan Download Format	48
Gambar 5.4 UI File Chooser	49
Gambar 5.5 UI Matriks Transportasi	49
Gambar 5.6 Keterangan Format.txt	50
Gambar 5.7 Format dalam angka.txt.....	50
Gambar 5.8 UI Matriks <i>Least Cost</i>	51
Gambar 5.9 UI Matriks <i>Stepping Stone</i> , Reset Table dan Efisiensi biaya.....	51

Gambar 5.10 UI Matriks <i>Stepping Stone</i> , Reset Table dan Efisiensi biaya	52
Gambar 5.11 Masukkan validasi java.....	54
Gambar 5.12 hasil perhitungan <i>least cost</i>	54
Gambar 5.13 hasil perhitungan <i>stepping stone</i>	55
Gambar 5.14 Hasil <i>Output java stepping stone loop</i>	56
Gambar 6.1 File Masukkan Input1.txt.....	60
Gambar 6.2 File Masukkan Input1.txt bagian 2 horizontal	60
Gambar 6.3 UI Matriks awal <i>least cost</i>	61
Gambar 6.4 UI Matriks <i>Least Cost</i> Bagian 1	62
Gambar 6.5 UI Matriks <i>Least Cost</i> Bagian 2.....	62
Gambar 6.6 UI Matriks <i>Least Cost</i> Bagian 3.....	63
Gambar 6.7 UI Matriks <i>Least Cost</i> Bagian 4.....	63
Gambar 6.8 UI Matriks <i>Stepping Stone</i> Bagian 1.....	65
Gambar 6.9 UI Matriks <i>Stepping Stone</i> Bagian 2.....	65
Gambar 6.10 UI Matriks <i>Stepping Stone</i> Bagian 3.....	66
Gambar 6.11 UI Matriks <i>Stepping Stone</i> Bagian 4.....	66
Gambar 6.12 Biaya Eksisting Bagian 1.....	71
Gambar 6.13 Biaya Eksisting Bagian 2.....	71
Gambar 7.1 Data realisasi Bag 1.....	79
Gambar 7.2 Data realisasi Bag 1 ke 2.....	80
Gambar 7.3 Data realisasi Bag 2.....	81
Gambar 7.4 Data realisasi Bag 2 ke 2.....	82
Gambar 7.5 Data realisasi Bag 3.....	83

Gambar 7.6 Data realisasi Bag 3 ke 2.....	84
Gambar 8.1 Data tabel pivot bagi 1.....	85
Gambar 8.2 Data tabel pivot bagi 2.....	86
Gambar 8.3 Data tabel pivot bagi 3.....	87
Gambar 8.4 Data tabel pivot bagi 4.....	88

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 4. 1 Justifikasi kategori Pivot Fields	24
Tabel 4.2 Tabel Depot Supply dan Kapasitasnya	26
Tabel 4.3 Tabel Rasio Jarak Tempuh.....	29
Tabel 4.4 Tabel Kondisi Proses Distribusi.....	32
Tabel 4.5 Matriks transportasi model matematis.....	32
Tabel 5.1 Tabel Perangkat Keras dan Lunak yang digunakan..	35
Tabel 5.2 Matriks transportasi validasi.....	53
Tabel 5.3 Matriks transportasi <i>least cost</i> validasi.....	53
Tabel 5.4 Matriks transportasi <i>stepping stone</i> validasi.....	55
Tabel 6.1 Alokasi <i>Least Cost</i>	67
Tabel 6.2 Alokasi <i>Stepping Stone</i>	69
Tabel 6.3 Biaya Eksisting.....	72
Tabel 6.4 Efisiensi Biaya.....	72

DAFTAR KODE PROGRAM

Kode Program 5.1 Algoritma <i>matrix</i>	36
Kode Program 5.2 Algoritma <i>least cost</i>	38
Kode Program 5.3 Algoritma <i>clone array</i>	38
Kode Program 5.4 Algoritma <i>mincost index</i>	39
Kode Program 5.5 Algoritma <i>stepping stone</i> bag 1.....	40
Kode Program 5.6 Algoritma <i>stepping stone</i> bag 2.....	40
Kode Program 5.7 Algoritma <i>degenerate case</i>	41
Kode Program 5.8 Algoritma penentuan <i>sel stepping stone</i>	42
Kode Program 5.9 Algoritma pengalokasian demand <i>pada least cost</i>	43
Kode Program 5.10 Algoritma penghitungan total cost.....	44

BAB I

PENDAHULUAN

Pada bab ini, akan dijelaskan mengenai latar belakang masalah yang menyebabkan studi kasus ini diangkat menjadi tugas akhir, perumusan masalah pengerjaan tugas akhir, batasan masalah pengerjaan tugas akhir, tujuan pengerjaan tugas akhir, manfaat yang dapat diambil dari hasil tugas akhir, serta relevansi hasil tugas akhir terhadap bidang keilmuan jurusan dan laboratorium.

1.1 Latar Belakang

Bahan Bakar Minyak (BBM) sangat kita perlukan dalam kehidupan sehari-hari. Karena sebagian besar transportasi yang kita gunakan menggunakan BBM. Jenis-jenis BBM yg dapat kita temui di Indonesia seperti minyak bensin, minyak solar, minyak tanah, dll. Setiap tahunnya konsumsi BBM meningkat, contoh pada tahun 2015 ke 2016 mengalami peningkatan sebesar 38,80%.[1] Meskipun demikian, tidak jarang terjadi kelangkaan BBM di daerah-daerah tertentu. Tanpa adanya BBM, mobilitas kita pasti terganggu bahkan bisa menurun secara drastis.

Salah satu perusahaan yang bergerak dibidang pendistribusian BBM adalah PT. Pertamina Patra Niaga. Perusahaan tersebut merupakan anak perusahaan dari PT Pertamina yang merupakan perusahaan utama di Indonesia sebagai penyedia BBM. PT. Pertamina Patra Niaga sendiri sudah melakukan Fuel Trading dari tahun 2004. Jadi perusahaan tersebut juga menjadi salah satu perusahaan yang bertanggung jawab dalam pendistribusian BBM yang masuk ke Indonesia. [3]

Penjualan utama dari PT. Pertamina Patra Niaga adalah bahan bakar diesel yang diberi nama produk HSD (Patra Diesel). HSD sendiri didistribusikan ke *bunker* lain dan ke industri industri lain yang membutuhkan sesuai *demand*.

Dari permasalahan di atas, distribusi mengambil peran penting dalam pemerataan penyaluran BBM. Masalah transportasi pun juga sering kita hadapi dalam proses pendistribusian barang. Dengan menggunakan transportasi yang tepat dan jalur ke destinasi yang tepat pula maka pendistribusian akan berjalan secara optimal.[2] Tak lepas dari itu, pendistribusian barang juga memerlukan biaya. Jika biaya tersebut bisa diminimalkan, pendistribusian barang juga bisa lebih merata. Tetapi juga perlu diperhatikan bagaimana cara meminimalkan biaya pendistribusian barang tersebut. Ketidaktepatan pengalokasian akan menambah biaya pendistribusian, dimana akan berimbas langsung pada kemampuan pendistribusian dan menyebabkan kerugian.

Untuk mendapatkan hasil yang optimal dalam pendistribusian barang, ada beberapa metode yang bisa digunakan dan dapat ditemukan dalam Riset Operasi. Di dalam Riset Operasi, masalah transportasi termasuk bagian dari *Linier Programming*. [4] Diharapkan dengan menggunakan metode yang termasuk dalam *Linier Programming* tersebut dapat mengoptimalkan biaya distribusi.

Dengan Metode *Least Cost* (LC) adalah salah satu metode transportasi yang paling cepat untuk dilakukan, tetapi hasilnya belum tentu optimal. Dalam LC diasumsikan tiap lokasi sumber dan lokasi tujuan diurutkan dari sisi kiri ke kanan dan dari atas ke bawah dalam peta data matriks. Cara penghitungan biaya transportasi dengan menggunakan metode LC sesuai dengan namanya dimulai dari matriks demand biaya terendah, kemudian bergerak mengisi demand-demand dengan biaya terendah lainya sesuai dengan kapasitas supply untuk memenuhi kapasitas *demand*. [6] Selanjutnya untuk hasil solusi yang lebih optimal, kemudian digunakan Metode *Stepping Stone* (SS).

PT. Pertamina Patra Niaga setiap bulannya mendapatkan ribuan demand dari perindustrian di Indonesia. Disini perlu diingat

juga bahwa penggunaan metode-metode diatas juga diperlukan langkah yang panjang dan cukup memakan waktu untuk menyelesaikannya. Dengan demikian akan lebih tepat untuk menambahkan bahasa pemrograman java untuk mengkalkulasi matriks metode-motode tersebut.

Berdasarkan uraian diatas, maka penulis mengajukan pengoptimalan biaya transportasi untuk distribusi BBM PT. Pertamina Patra Niaga menggunakan metode LC dan metode SS dengan bahasa pemrograman Java mengambil studi kasus di Region Jatim Balinus dan khusus untuk distribusi HSD yang didistribusikan ke *bunker*.

1.2 Perumusan Masalah

Berdasarkan uraian pada latar belakang di atas, berikut adalah perumusan masalah yang dijadikan acuan dalam pengerjaan tugas akhir ini:

1. Bagaimana langkah penggunaan Metode LC dan Metode SS dengan bahasa pemrograman java pada PT. Pertamina Patra Niaga Region Jatim Balinus?
2. Apakah dengan menggunakan metode tersebut biaya distribusi lebih optimal?

1.3 Batasan Masalah

Untuk menghasilkan pembahasan yang jelas dan terarah dalam pengerjaan tugas akhir ini, ada beberapa batasan masalah yang harus diperhatikan, yaitu sebagai berikut:

1. Pengoptimalan biaya transportasi meliputi PT. Pertamina Patra Niaga Region Jatim Balinus
2. Produk yang digunakan hanya HSD yang di distribusikan dari *Bunker*
3. Data yang digunakan adalah data realisasi Region Jatim Balinus Desember tahun 2016

1.4 Tujuan Penelitian

Tujuan dari pengerjaan tugas akhir ini adalah:

1. Menerapkan Metode LC dan Metode SS dalam distribusi HSD di Region Jatim Balinus dengan bahasa pemrograman java
2. Mengoptimalkan biaya transportasi untuk distribusi HSD di Region Jatim Balinus
3. Mengetahui perbandingan biaya sebelum menggunakan dan sesudah menggunakan metode tersebut

1.5 Manfaat Penelitian

Manfaat yang diberikan dengan adanya tugas akhir ini adalah sebagai berikut:

1. Bagian Operational PT. Pertamina Patra Niaga Jatim Balinus mengetahui penerapan metode LC dan SS pada pendistribusian BBM dan penggunaan aplikasi berbasis java yang telah dibuat
2. Mendapatkan masukan rekomendasi untuk mengoptimalkan biaya transportasi untuk distribusi BBM

1.6 Relevansi

Tugas akhir ini berkaitan dengan mata kuliah Riset Operasi dan masuk ke dalam bidang keilmuan laboratorium Rekayasa Data dan Intelegensi Bisnis (RDIB).

BAB II

TINJAUAN PUSTAKA

Sebelum melakukan penelitian tugas akhir, penulis melakukan tinjauan pustaka terhadap tulisan dari beberapa penelitian sebelumnya yang sesuai dengan tema yang diambil. Dan dibawah ini merupakan rincian hasil yang didapatkan :

2. 1 Penelitian Sebelumnya

Penelitian yang dijadikan acuan dalam pengerjaan tugas akhir ini adalah:

1. Implementasi Pengotimalan Biaya Transportasi dengan NorthWest Corner Method (NWCM) dan *Stepping Stone* Method (SSM) Untuk Distribusi Raskin pada Perum Bulog Sub Divre Semarang oleh Nur Laely Fatimah. Pada penelitian ini dilakukan uji optimasi dengan menggunakan metode tersebut dimana QM for windows menunjukkan hasil yang sama, walaupun masih terdapat selisih biaya yang sangat sedikit, namun ada kesamaan secara signifikan sehingga program NWCM dan SSM dikatakan valid dengan pengujian program standar. [8]
2. Analisis Penerapan Model Transportasi Distribusi dengan Menggunakan NWCM dan SSM pada Harian Tribun Makassar oleh Acmad Hariyono. Pada penelitian ini pengimplementasian metode tersebut dapat menghemat biaya transportasi distribusi.[9]
3. Analisis Perbandingan Pengiriman Barang Menggunakan Metode Transportasi (Studi Kasus di PT. Arta Boga Jakarta) oleh Miptahudin. Pada penelitian ini dilakukan perbandingan metode transportasi yang digunakan, dimana penggunaan metode SS sangat berpengaruh untuk menentukan solusi optimal.[10]

2.2 Dasar Teori

2.2.1 Vehicle Routing Problem

Dantzig dan Ramzer memperkenalkan *Vehicle Routing Problem (VRP)* untuk menyelesaikan masalah transportasi. Dimana ada sebuah depot awal dan masing-masing tujuan yang memiliki demand yang harus dipenuhi. Dimana demand yang ada dapat berbeda-beda dari satu tujuan ke tujuan lain.[11]

Tujuan dari *Vehicle Routing Problem (VRP)* sendiri adalah meminimalkan biaya distribusi yang dikeluarkan sehubungan dengan sumber daya yang ada. Dimana dari depot yang ada harus menyampaikan demand yang diinginkan ke tujuan-tujuan tersebut.

Penggunaan metode VRP harus mempertimbangkan sumber daya dan demand yang ada. Misalkan tujuan demand harus lebih dari 1 dan jumlah dari demand harus berbeda-beda. Syarat-syarat dasar dari VRP yaitu:

1. Rute berawal dari depot menuju tujuan demand dan berakhir ke depot kembali.
2. Setiap tujuan harus diselesaikan 1 kendaraan.
3. Jumlah total demand tidak boleh melebihi total supply yang ada.
4. Waktu tempuh selama distribusi tidak boleh melebihi batasan yang ditetapkan.
5. Dengan tujuan meminimumkan biaya.

Ada beberapa faktor yang menentukan dalam VRP yaitu *vehicle*, *customer*, *depot*, *driver*, dan faktor-faktor lain :

Vehicle memiliki beberapa karakteristik sebagai berikut :

1. Tiap kendaraan mempunyai kapasitas angkut masing-masing sesuai realita.

2. Tiap kendaraan harus mempunyai total waktu kerja pasti yang diperhitungkan dari awal berangkatnya dari depot sampai tujuan demand hingga kembalinya lagi ke depot.
3. Tiap kendaraan mempunyai waktu yang tidak dapat pasti diperhitungkan misalnya: istirahat pengemudi, mengisi bahan bakar dll.
4. Tiap kendaraan mempunyai perhitungan ongkos pengiriman.

Customer memiliki karakteristik sebagai berikut :

1. Tiap *customer* mempunyai jumlah demand yang bermacam-macam, baik dalam jumlah yang pasti maupun tidak.
2. Tiap *customer* mempunyai periode waktu tertentu dalam penerimaan demand yang diinginkan. Misalnya: batas deadline penerimaan, jendela waktu penerimaan dll.

Depot merupakan titik awal keberangkatan transportasi menuju ke demand, dimana kendaraan memuat barang yang nantinya dikirimkan ke tujuan yang memiliki demand atas barang tersebut.

Driver merupakan pengemudi/penggerak dari kendaraan yang digunakan untuk mendistribusikan, dimana harus ditentukan pekerjaan dan batasan yang boleh dilakukan oleh driver tersebut. Seperti total jam istirahat, tujuan dan deadline distribusi.

Beberapa faktor lain yang ada dalam VRP:

1. *Multiple trips* : perjalanan dilakukan berulang-ulang oleh kendaraan yang sama baik tujuan yang sama ataupun berbeda dalam satu hari.
2. Perjalanan dilakukan melebihi batas waktu harian 24 jam.
3. Penggolongan tipe kendaraan berdasarkan volume/kapasitas kendaraan serta fungsionalitasnya

4. Terdapat lebih dari 1 depot, dimana kendaraan bisa memulai proses distribusinya dari depot yang berbeda setelah ditentukan.

Berdasarkan teori VRP diatas, secara khusus bertujuan untuk meminimalkan biaya transportasi berdasarkan rute yang diambil dan mengacu pada ketepatan waktu. Juga perlu dipertimbangkan kesesuaian metode yang digunakan untuk menyelesaikan masalah.

2.2.2 Riset Operasi

Riset Operasi (RO) secara garis besar mempunyai definisi yang berbeda-beda menurut beberapa ahli. Contohnya:

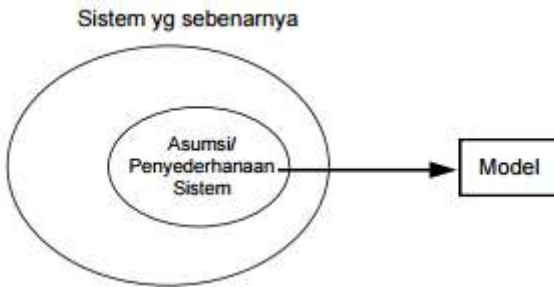
1. Operational Research Society of Great Britain: penerapan metode-metode ilmiah terhadap masalah-masalah rumit yang muncul dalam pengarahannya dan pengelolaan dari suatu sistem besar manusia, mesin, bahan dan uang dalam industri, bisnis, pemerintahan dan pertahanan.
2. Operation Research Society of America: berkaitan dengan menentukan pilihan secara ilmiah bagaimana merancang dan menjalankan sistem manusia-mesin secara terbaik, biasanya membutuhkan alokasi sumber daya yang langka.
3. T.L. Saaty: Riset operasi adalah seni memberikan jawaban buruk terhadap masalah-masalah, yang jika tidak, memiliki jawaban yang lebih buruk.
4. Hamdi A. Taha: pendekatan dalam pengambilan keputusan yang ditandai dengan penggunaan pengetahuan ilmiah melalui usaha kelompok antar disiplin yang bertujuan menentukan penggunaan terbaik sumber daya yang terbatas.
5. Churchman, Ackoff, dan Arnoff: penerapan metode-metode, teknik-teknik, dan alat-alat terhadap masalah-masalah yang

menyangkut operasi-operasi dari sistem-sistem, sedemikian rupa sehingga memberikan penyelesaian optimal. [12]

Menurut penulis, RO berhubungan dengan prinsip optimisasi, yaitu bagaimana cara menggunakan sumber daya (waktu, tenaga, biaya, dll) untuk mengoptimalkan hasil. Mengoptimalkan hasil dapat berarti meminimumkan sesuatu yang dikeluarkan atau memaksimumkan sesuatu yang didapatkan. Hal ini sesuai dengan studi kasus yang dibahas, karena bertujuan untuk meminimalkan biaya distribusi yang dikeluarkan. Beberapa contoh kasus dalam kehidupan sehari-hari yang dapat di selesaikan RO:

1. Pemilihan rute ke tempat tujuan, dimana akan berimbas pada sumber daya/ tenaga yang dikeluarkan.
2. Bahan baku pembuatan barang, dimana pemilihan bahan yang tepat guna dapat meminimalkan biaya produksi.
3. Penugasan setiap pegawai yang tepat akan meminimumkan penggunaan sumber daya dan peningkatan pendapatan.

Riset Operasi mencari keputusan/hasil terbaik pada penyelesaian suatu masalah yang memenuhi beberapa kondisi yang ditentukan. Dalam prosesnya Riset Operasi berhubungan dengan Model. Model adalah interaksi/hubungan antara variabel-variabel yang mempengaruhi sistemnya.



Gambar 2.1 Permodelan OR

Kompleksnya sistem yang dipelajari akan membuat penyelesaian masalah menjadi sulit. Untuk itu perlu untuk mereduksi "dimensi" sistem sehingga model (tiruan sistem) dapat dibuat seperti gambar 1.1. Biasanya, diantara sekian banyak faktor/variabel yang mempengaruhi sistem, hanya beberapa diantaranya saja yang penting dan memberi efek yang nyata terhadap sistem. Untuk menyederhanakan sistem, faktor-faktor yang kurang penting dibuang/diasumsikan.

Dalam Riset Operasi, pembuatan model melibatkan 3 komponen dasar yang penting :

1. Variabel Keputusan yaitu merupakan hasil yang dicari dalam permodelan/ dimana keputusan yang harus dibuat yang nantinya akan membuat perbedaan solusi yang ada.
2. Tujuan dalam riset operasi yaitu meminimalkan atau memaksimalkan dari penghitungan model matematis yang ditentukan.
3. Constraint adalah ketentuan yang membatasi variabel keputusan yang boleh digunakan.

2.2.3 Linear Programming

Linear programming (LP) adalah suatu cara untuk menggunakan sumber daya yang sifatnya terbatas untuk mencapai tujuan. Dalam studi kasus ini sumber daya yang ada berupa jumlah supply, jumlah depot dan biaya yang dikeluarkan untuk distribusi. Karakteristik LP sebagai berikut:

1. Penyelesaian masalah berupa meminimalkan sumber daya atau memaksimalkan pendapatan.
2. Batasan mempengaruhi fleksibilitas keputusan yang dibuat
3. Terdapat lebih dari 1 solusi
4. Hubungan matematis bersifat linear Secara teknis, ada lima syarat tambahan dari permasalahan linear programming yang harus diperhatikan yang merupakan asumsi dasar, yaitu:
 1. *Certainty* (kepastian). Maksudnya adalah fungsi tujuan dan fungsi kendala sudah diketahui dengan pasti dan tidak berubah selama periode analisa.
 2. *Proportionality* (proporsionalitas). Yaitu adanya proporsionalitas dalam fungsi tujuan dan fungsi kendala.
 3. *Additivity* (penambahan). Artinya aktivitas total sama dengan penjumlahan aktivitas individu.
 4. *Divisibility* (bisa dibagi-bagi). Maksudnya solusi tidak harus merupakan bilangan integer (bilangan bulat), tetapi bisa juga berupa pecahan.
 5. *Non-negative variable* (variabel tidak negatif). Artinya bahwa semua nilai jawaban atau variabel tidak negatif.[14]

Dengan menggunakan salah satu metode LP yaitu model transportasi, penulis mencoba meminimalkan biaya distribusi.

2.2.4 Metode Transportasi

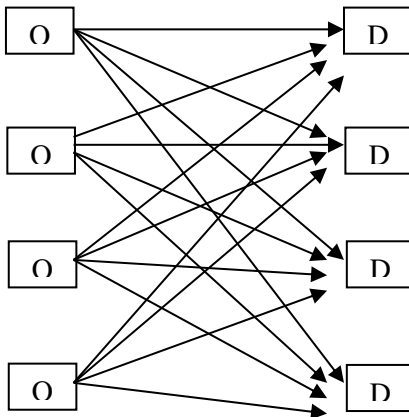
Metode Transportasi merupakan suatu metode yang digunakan untuk mengatur distribusi dari tempat asal ke tujuan dimana tempat asal dan tujuan harus lebih dari 1 dan terdapat perbedaan biaya yang harus dikeluarkan dari masing-masing

distribusi dari tempat asal ke tujuan yang diinginkan berdasarkan demand di tempat tujuan. Dengan tujuan meminimalkan biaya yang digunakan dalam distribusi tersebut [4].

Model transportasi mempunyai karakteristik dan syarat-syarat sebagai berikut:

1. Adanya pengalokasian barang yang dipindahkan dari sumber ke tempat tujuan berdasarkan demand
2. 1 Sumber dapat mendistribusikan ke beberapa/semua tujuan.
3. 1 Tujuan dapat menerima distribusi dari beberapa sumber.
4. Masing-Masing sumber dan tujuan mempunyai kapasitas, dimana sumber mempunyai kapasitas pengiriman (supply) dan tujuan mempunyai kapasitas permintaan (demand).
5. Alokasi distribusi pada masing-masing sumber dan tujuan tidak boleh melebihi kapasitasnya.

Pada gambar dibawah diterangkan bagaimana cara distribusi model transportasi dari sumber (*origin*) ke tujuan (*destination*) dimana setiap sumber bisa mendistribusikan ke setiap tujuan begitu pula sebaliknya.



Gambar 2.2 Distribusi Model Transportasi

Dalam model transportasi terdapat 3 metode yang dapat digunakan untuk menentukan solusi awal, yaitu :

1. Metode North West Corner (Metode Sudut Barat Laut)
2. Metode *Least Cost* (Metode Biaya Terkecil)
3. Metode VAM (Vogel's Approximation Method)

Setelah itu untuk menentukan solusi akhir yang optimal dapat digunakan:

1. Metode Modified Distribution (MODI)
2. Metode *Stepping Stone*

Disini penulis menggunakan metode *Least cost* dan untuk solusi akhir yang optimal akan dilanjutkan menggunakan metode *Stepping Stone*.

2.2.5 Formulasi Model Matematis

Mencari $X_{ij} \geq 0$ ($i = 1, 2, \dots, m$; $j = 1, \dots, n$) yang meminimalkan fungsi sasaran (ongkos angkut total).

$$f = \sum_{j=1}^n \sum_{i=1}^m C_{ij} X_{ij} ,$$

dengan kendala-kendala (*constraint*) :

1. $\sum_{j=1}^n X_{ij} \leq b_i \quad (i = 1, 2, \dots, m), \text{ dan}$
2. $\sum_{i=1}^m X_{ij} \geq a_j \quad (j = 1, 2, \dots, n)$

Data masalah transportasi dituangkan dalam table berikut :

Tabel 2.1 Matrix Transportasi

Destination Origin	D ₁	D ₂	...	D _n	Supply b _i
O ₁	$\begin{matrix} & c_{11} \\ x_{11} & \end{matrix}$	$\begin{matrix} & c_{12} \\ x_{12} & \end{matrix}$...	$\begin{matrix} & c_{1n} \\ x_{1n} & \end{matrix}$	b ₁
O ₂	$\begin{matrix} & c_{21} \\ x_{21} & \end{matrix}$	$\begin{matrix} & c_{22} \\ x_{22} & \end{matrix}$...	$\begin{matrix} & c_{2n} \\ x_{2n} & \end{matrix}$	b ₂
...
O _m	$\begin{matrix} & c_{m1} \\ x_{m1} & \end{matrix}$	$\begin{matrix} & c_{m2} \\ x_{m2} & \end{matrix}$...	$\begin{matrix} & c_{mn} \\ x_{mn} & \end{matrix}$	b _m
Demand a _j	$\begin{matrix} a_1 \\ \\ \\ \end{matrix}$	$\begin{matrix} a_2 \\ \\ \\ \end{matrix}$...	$\begin{matrix} a_n \\ \\ \\ \end{matrix}$	$\begin{matrix} \sum b_i \\ \sum a_j \end{matrix}$

Keterangan Tabel 2.1:

1. O_i = Sumber (*origin*) ke – i (i = 1, 2, . . . , m)
2. D_j = Tujuan (*destination*) ke – j (j = 1, 2, . . . , n)
3. b_i = Supply maksimum pada O_i
4. a_j = Demand minimum pada D_j
5. C_{ij} = Ongkos angkut satuan pada jalur O_i ke D_j
6. X_{ij} = Banyaknya unit komoditi yang diangkut dari O_i ke D_j (alokasi).[15]

Diasumsikan :

1. Linieritas, i.e. biaya angkut berbanding lurus (proporsional) dengan banyaknya komoditi yang diangkut dari *origin* ke *destination*.
2. Hanya ada satu jenis komoditi yang diangkut

Asumsi 1 berakibat masalah transportasi termasuk dalam kategori masalah program linear, Sehingga cara menyelesaikannya bisa memanfaatkan metode yang sudah lasim dikenal, seperti yang akan dijabarkan kemudian.

Asumsi 2 berakibat setiap *destination* bisa menerima kiriman dari setiap *origin*.

2.2.6 *Least Cost*

Metode ini dikerjakan sebagai berikut:

- a) Mengalokasian $\mathbf{b_i}$ pada $\mathbf{X_{ij}}$ berdasarkan $\mathbf{a_j}$ pada kolom dengan $\mathbf{C_{ij}}$ terendah
- b) Apabila ada beberapa kolom $\mathbf{C_{ij}}$ terendah, dapat dipilih secara acak tidak berdasarkan apapun.
- c) Langkah **a) dan b)** dikerjakan terus menerus sampai dengan terpenuhinya $\mathbf{a_j}$, abaikan apabila pada kolom $\mathbf{C_{ij}}$ terendah sudah memenuhi $\mathbf{b_i}$ ataupun $\mathbf{a_j}$ [16]

2.2.7 *Stepping Stone*

Metode *Stepping Stone* (SS) dilakukan dengan langkah-langkah berikut:

1. Untuk setiap kotak kosong $\mathbf{X_{ij}}$ dicari lintasan horisontal & vertikal (tertutup/loop) melewati kotak-kotak yang sudah isi. Loop ini selalu bisa diperoleh, karena kita sudah mempunyai $m+n-1$ kotak isi. Sebagai gambaran misalkan kita mempunyai kotak kosong yang mempunyai lintasan tertutup

$$X_{13} \rightarrow X_{14} \rightarrow X_{34} \rightarrow X_{33} \rightarrow X_{13},$$
2. maka “*opportunity Cost*” c_{13}^* didefinisikan sebagai :

$$c_{13}^* = -\Delta f_{13},$$
 di mana $\Delta f_{13} = c_{13} - c_{14} + c_{34} - c_{33}.$
3. Hitunglah *opportunity cost* c_{ij} untuk setiap kotak kosong $X_{ij}.$

4. Solusi sudah optimal bila dan hanya bila /jika opportunity cost $c_{ij}^* \leq 0$, untuk semua kotak kosong x_{ij} .
5. Solusi belum optimal jika terdapat opportunity cost $c_{ij}^* > 0$, untuk suatu kotak kosong x_{ij} jika ini terjadi, maka langkah selanjutnya ulangi metode SS.[16]

2.2.8 Java Programming

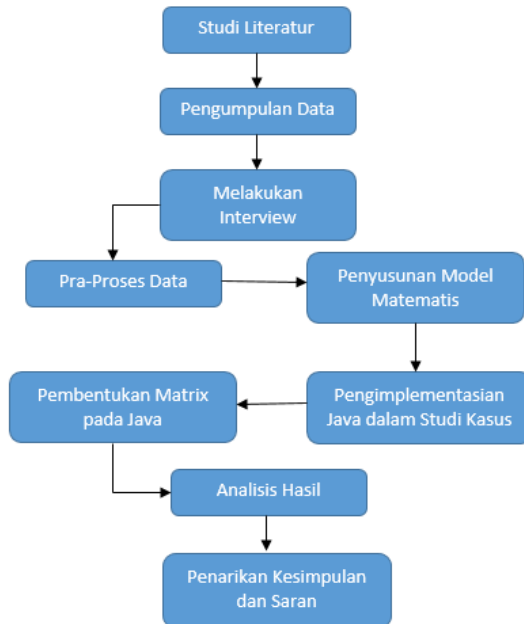
Java merupakan bahasa berorientasi objek (OOP) yaitu cara ampuh dalam pengorganisasian dan pengembangan perangkat lunak. Pada OOP, program komputer sebagai kelompok objek yang saling berinteraksi. Deskripsi ringkas OOP adalah mengorganisasikan program sebagai kumpulan komponen, disebut objek. Objek-objek ini ada secara independen, mempunyai aturan-aturan berkomunikasi dengan objek lain dan untuk memerintahkan objek lain guna meminta informasi tertentu atau meminta objek lain mengerjakan sesuatu. Kelas bertindak sebagai modul sekaligus tipe. Sebagai tipe maka pada saat jalan, program menciptakan objek-objek yang merupakan instan-instan kelas. Kelas dapat mewarisi kelas lain singkatnya seperti itu.

Disini Penulis akan menggunakan bahasa pemrogramman Java untuk membantu dan mempermudah dalam penghitungan metode transportasi diatas.

BAB III

METODOLOGI PENELITIAN

Pada bab ini akan dijelaskan mengenai tahapan-tahapan sistematis yang dilakukan dalam pengerjaan tugas akhir ini. Tahapan-tahapan tersebut digunakan sebagai panduan agar pengerjaan tugas akhir ini terlaksana secara terarah dan terstruktur. Diagram alur dari metodologi yang digunakan dalam tugas akhir ini ditunjukkan pada Gambar 3.1. Penjelasan mengenai masing-masing tahapan proses dalam pengerjaan tugas akhir ini berdasarkan diagram alur tersebut adalah sebagai berikut.



Gambar 3.1 Diagram Alur Pengerjaan Tugas Akhir

3.1 Studi Literatur

Tahapan studi literatur mempunyai tujuan agar dapat menemukan permasalahan dalam pengerjaan tugas akhir ini dan memiliki landasan teori maupun referensi metode yang sesuai dengan topik pembahasan. Beberapa sumber yang dijadikan acuan pada tahapan studi literatur ini didapatkan dari buku, jurnal ilmiah, artikel maupun tugas akhir yang dipublikasikan. Keluaran dari proses ini adalah kerangka konseptual model dari metode yang digunakan dalam pengerjaan tugas akhir ini.

3.2 Pengumpulan Data

Tahapan selanjutnya yaitu mengumpulkan data yang sesuai dengan permasalahan beserta batasan masalah yang telah ditentukan. Data diambil secara langsung dan melakukan wawancara dengan pihak dari PT. Pertamina Patra Niaga. Contoh bentuk awal data yang didapatkan ada pada Gambar 3.2. Proses wawancara dibutuhkan untuk mendukung informasi dari data-data yang telah diberikan. Data yang diharuskan ada dalam pengerjaan tugas akhir ini adalah sebagai berikut :

- a. Transaksi demand HSD Desember 2016
- b. Jumlah Suplier/*Bunker*
- c. Beban dan alur Distribusi

Gambar 3. 2 Contoh Data Realisasi HSD

3.3 Interview

Interview dilakukan dengan manajer *inventory*, *sales* dan *operasional* dari PT. Pertamina Patra Niaga Jatim Balinus ini dilakukan untuk mengambil data-data yang dibutuhkan secara valid dan terpercaya. Disini penulis mendapatkan Transaksi demand HSD Desember 2016, jumlah *bunker* dan rumus biaya pengiriman.

3.4 Pra-proses Data

Setelah mendapatkan data, maka perlu dilakukan pra-proses data untuk memilah dan menggabungkan kembali data-data yang akan dibutuhkan pada pengerjaan tugas akhir ini. Dalam pra-proses data, perlu dilakukan penyaringan data sesuai dengan batasan masalah yang dipakai penulis. Dengan adanya pra-proses data, maka akan memudahkan dalam pengerjaan tugas akhir.

3.5 Penyusunan Model Matematis

Dari data-data yang telah di pra-proses diatas, kemudian selanjutnya dikonversikan menjadi sebuah model matematis. Dimana model matematis memiliki *constraint*, fungsi tujuan dan *variable* keputusan.

Dalam hal penyusunan model matematis yang akan dibentuk megacu pada bab 2.2.6 dimana model matematis tersebut bertumpu pada matrix transportasi.

3.6 Pengimplementasian Java pada Studi Kasus

Disini penulis akan membuat aplikasi dengan bahasa pemrograman Java. Jadi penghitungan menggunakan metode *least cost* dan dilanjutkan dengan metode *stepping stone* akan berjalan dengan sendirinya hingga menemukan hasil akhir.

Hanya dengan memasukkan data jumlah *supplier*, *demand* dan masing-masing biaya yang dibutuhkan dari *supplier* ke tujuan pendistribusian.

3.7 Pembentukan Matriks Transportasi pada Java

Pembentukan Matriks Transportasi pada Java dapat dilakukan setelah pengimplementasian java pada studik kasus sebelumnya telah dan selesai dilakukan. Karena pembentukan matrix akan berfokus pada formatting dan GUI.

Dengan demikian dapat dilihat tampilan informatif sehubungan dengan pengimplementasian metode pada studi kasus yang dikerjakan penulis.

3.8 Analisis Hasil

Pada proses ini akan dilakukan anilisis dengan cara membandingkan hasil yang didapat dari data sales realisasi Jatim Balinus periode 2016 khusus pada bulan Desember dan hasil yang didapat dari

pengimplementasian Java yang dilakukan pada studi kasus ini.

3.9 Penarikan Kesimpulan dan Saran

Setelah melalui proses analisis hasil, yaitu dengan membandingkan data realisasi dan hasil pengimplementasian java. Penulis akan melakukan penarikan kesimpulan apakah dengan mengimplementasikan metode *least cost* dan dilanjutkan dengan *stepping stone* menggunakan bahasa pemrograman java lebih optimal atau tidak.

Sesuai dengan kesimpulan tersebut penulis juga akan membuat saran untuk PT. Pertamina Patra Niaga Region Jatim Balinus dan diharapkan dapat berguna dan bermanfaat serta sesuai harapan.

Halaman ini sengaja dikosongkan

BAB IV PERANCANGAN

Pada bab ini, akan dijelaskan mengenai rancangan bagaimana penelitian akan dilakukan. Terdiri dari penjelasan mengenai bagaimana pengumpulan data yang akan digunakan pada penelitian tugas akhir dan pra-proses data yang akan digunakan.

4.1 Pra-Proses data

Pada data realisasi yang didapatkan, berisi rincian transaksi dari bulan januari sampai dengan Desember 2016. Sementara itu data yang dimaksud sesuai batasan hanya memperhitungkan data transaksi pada bulan Desember 2016. Ada pula material yang digunakan hanya di fokuskan kepada HSD.

Maka dari itu, pra-proses data perlu dilakukan agar data yang digunakan sesuai dengan batasan yang dibuat penulis. Dengan menggunakan Pivot table pada excel, penulis bisa memilah-milah data mana yang valid untuk digunakan.

Dalam penggunaan pivot tabel sebagai pra-prosesing, banyak hal yang harus diperhatikan diantaranya:

1. *Pivot Fields*
2. *Filter*
3. *Columns*
4. *Rows*
5. *Values*

Menentukan 5 kategori diatas haruslah tepat agar data yang didapat benar-benar valid. Langkah pertama yang dilakukan adalah memilih *Pivot Fields* mana yang akan dipergunakan dalam studi kasus ini.

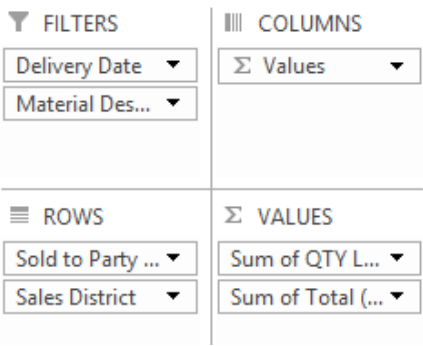
Pivot Fields atau kategori penting yang diambil dalam kasus ini ada 6 terdiri dari *Delivery Date*, *Material Description*, *Sold to Party Name*, *Sales District*, *QTY LTR*, dan *Total (IDR)*.

Berikut justifikasi mengapa kategori diatas yang diambil pada tabel 4.1:

Tabel 4. 1 Justifikasi kategori *Pivot Fields*

Kategori	Justifikasi
<i>Delivery Date</i>	Data yang diambil hanya pada pengiriman Desember 2016
<i>Material Description</i>	Material yang di fokuskan hanya pada HSD-PTM
<i>Sold to Party Name</i>	Sebagai pemilah dari <i>destination</i> pada table matrix, karena ada pihak dengan destinasi sama tetapi pembuat <i>demand</i> berbeda
<i>Sales District</i>	Sebagai <i>destination</i> pada table matrix transportasi
QTY LTR	Sebagai jumlah <i>demand</i> yang diminta
Total (IDR)	Diambil hanya untuk kepentingan Analisis hasil

Setelah mendapatkan kategori-kategori diatas, dilanjutkan dengan memasukkan kategori-kategori tersebut kedalam x dan y pivot table. Kategori *Delivery Date* dan *Material Description* masuk pada *Filters*, kategori *Sold to Party Name* dan *Sales District* masuk kedalam *Rows*, kategori QTY LTR dan Total (IDR) masuk kedalam *Values* dimana nanti akan menampilkan jumlah dari masing masing kategori dan yang terakhir masukkan *Values* pada *Columns*. Untuk lebih detailnya dapat dilihat pada gambar 4.1



Gambar 4. 1 Contoh x dan y Pivot Tabel

Setelah melakukan langkah tersebut nantinya akan terlihat seperti gambar 4.2

Material Description	PATRA DIESEL/HSD - PTM
Row Labels	Sum of QTY LTR
Kab. Badung	1688480
Kab. Bangkalan	506000
Kab. Banyuwangi	2181000
Kab. Batu	5000
Kab. Blora	224000
Kab. Bojonegoro	103000
Kab. Bondowoso	8000
Kab. Boyolali	5000
Kab. Gianyar	46750
Kab. Gresik	13206000
Kab. Jember	273000
Kab. Jembrana	209000
Kab. Jombang	1051000
Kab. Karangasem	33000
Kab. Kediri	245000
Kab. Lamongan	382000
Kab. Lumajang	358000
Kab. Madiun	1328000
Kab. Magetan	18000
Kab. Malang	238000

Gambar 4. 2 Contoh hasil Pivot Tabel

Data yang ditampilkan diatas sudah melalui *filter Delivery Date* yang ditampilkan hanya pada bulan Desember 2016 dan *filter Material Description* yang hanya menampilkan HSD-PTM sesuai batasan masalah penulis.

4.2 Data yang digunakan

Data merupakan salah satu hal yang paling utama dalam pengerjaan tugas akhir ini karena data akan dijadikan input dalam proses. Terdapat tiga kelompok data yang diperlukan, yaitu data *supplier*, data *demand* dan data cara penghitungan beban biaya perliter. Berikut penjelasan dari masing-masing kelompok data :

4.2.1 Data *Supplier*

Dalam penghitungan Matriks transportasi diperlukan data *supplier* sebagai *origin* atau awal pendistribusian. Data *supplier* terdiri atas Nama *supplier* yang menerangkan letak dari *supplier* itu sendiri dan kapasitas yang dimilikinya.

Pada PT. Pertamina Patra Niaga Region Jatim Balinus terdapat 9 *Bunker Supply* yang dijabarkan pada table 4.2 dan sudah diurutkan berdasarkan letak dari Timur ke Barat

Tabel 4.2 Tabel Depot *Supply* dan Kapasitasnya

Letak Depot	Kapasitas
Ampenan	400KL
Badas	400KL
Atapupu	400KL
Manggis	800KL
Benoa	800KL
Banyuwangi	1500KL
Gresik	1200KL
Surabaya	1500KL
Cepu	55000KL

4.2.2 Data *Demand*

Data demand diambil dari data yang telah di pra-proses sebelumnya. Data yang telah di pra-proses juga sudah sesuai kebutuhan dari matrix transportasi.

Jadi dari data pra-proses diatas didapatkan 41 perusahaan yang melakukan permintaan pembelian dan total 41 tujuan distribusi/ *destination* berupa Depot di yang terletak di suatu Kabupaten/Kota.

PT. ADI PUTRA NARASI	290000
PT. AIRLANGGA PUTRA MANDIRI	65000
PT. ANTARIKSA LAKSANA	353000
PT. ARNOV ENERGI	20000
PT. ASLI MOJOPAHIT	10000
PT. BAHARI NUSANTARA	7000
PT. BUANA SURYA SEMESTA	29000
PT. CAHAYA PUTRA JAYA BALI	235000
PT. CITRA SEMESTA ENERGY	119000
PT. COCA COLA BOTTLING INDONESIA	15000
PT. ENERGI NIAGA UTAMA	1122000
PT. GREENFIELDS INDONESIA	21000
PT. INDOLINE INCOMEKITA	23000
PT. INTIM PERKASA	640000
PT. JAGAD NUSANTARA ENERGI	48000
PT. JAYADI SAMUDERA LINES	875000
PT. KEDAWUNG SUBUR	16000
PT. LAROS PETROLEUM	41000
PT. LIANINTI ABADI	210000
PT. NUSA CEMERLANG ENERGI	2500
PT. NUSANTARA NURAGA	21000
PT. PAITON ENERGY	1128000
PT. PATRA ANDALAS SUKSES	50000
PT. PATRA LOGISTIK	80000

Gambar 4. 3 Perusahaan yang melakukan permintaan pembelian/*demand*

Dari perusahaan-perusahaan pada gambar 4.3 dipilah lagi untuk menunjukkan tujuan/ *destination* yang akan dihitung dalam matrix transportasi.

PT. ADI PUTRA NARASI	290000
Kab. Sumbawa	100000
Kodya Denpasar	180000
Kodya Mataram	10000
PT. AIRLANGGA PUTRA MANDIRI	65000
PT. ANTARIKSA LAKSANA	353000
Kab. Gresik	56000
Kab. Lamongan	8000
Kab. Nganjuk	5000
Kab. Ponorogo	5000
Kab. Sidoarjo	245000
Kab. Trenggalek	4000
Kab. Tuban	8000
Kodya Mojokerto	8000
Kodya Pasuruan	1000
Kodya Surabaya	13000
PT. ARNOV ENERGI	20000
Kab. Sumbawa Barat	5000
Kab. Tabanan	10000
Kodya Surabaya	5000
PT. ASLI MOJOPAHIT	10000
Kab. Jembrana	5000
Kodya Denpasar	5000

Gambar 4. 4 Contoh *destination* dan *demand* dari setiap perusahaan

Dengan demikian dapat dihitung biaya beban transportasi perliter dari *demand* ke masing-masing *destination*.

4.2.3 Data penghitungan beban biaya distribusi

Setiap kegiatan distribusi akan dikenakan beban biaya sesuai *demand* yang diminta per *destination*. Dalam studi kasus ini beban biaya yang ditanggung berdasarkan jarak *origin* ke *destination*, muatan dalam liter dan aspek-aspek yang lain. Perlu diperhatikan pula adanya kebijakan rasio jarak tempuh yang ditetapkan PT. Pertamina Patra Niaga. Ada 3 rasio jarak tempuh yang ditetapkan pada Tabel 4.3, jarak tempuh aktual ditentukan dari jarak yang didapatkan pada *Google Map*

Tabel 4.3 Tabel Rasio Jarak Tempuh

Rasio Jarak Tempuh	Jarak Tempuh Aktual	Beban Biaya Distribusi
Jarak Tempuh Maksimum	Jarak tempuh ≥ 200 km	Tarif perliter ditentukan dari harga HSD perliter + 50 rupiah/liter
Jarak Tempuh Normal	Jarak tempuh > 100 km dan Jarak tempuh < 200 km	Tarif perliter ditentukan dari rumus tarif perliter dengan rasio jarak tempuh sesuai jarak aktual
Jarak Tempuh Minimum	Jarak tempuh < 100 km	Tarif perliter ditentukan dari rumus tarif perliter dengan rasio jarak tempuh = 100

Berikut merupakan rumus yang digunakan PT Patra Niaga Region Jatim Balinus untuk menghitung biaya distribusi:

Rumus Tarif Perliter :

- a. *Liter Own Use* = $\frac{\text{Rasio Jarak Tempuh} \times 2}{\text{Rasio MT}}$
- b. *BBM Own Use* = *LOU* x *Harga BBM Solar*
- c. *Tarif Vendor* = $50 \times \text{Rasio HSDPTM}$
- d. *Total Jumlah* = *BOU* + *Tarif Vendor*
- e. *Grand Total* = *Total Jumlah* + 20% *Total Jumlah*
- f. *Labours Supply* = $\frac{\text{Grand Total}}{\text{Kapasitas MT}}$
- g. *Tarif PerLiter* = *Labours Supply* x 20%

Keterangan:

- a. Rasio Jarak Tempuh seperti yang telah dijabarkan pada tabel 4.3

- b. Rasio MT = 5,4
- c. Harga BBM Solar Desember 2016 @Rp 9.300
- d. Rasio HSDPTM = 5000
- e. Kapasitas MT = 5000

Contoh penghitungan biaya perliter dengan rumus diatas sesuai rasio jarak tempuh:

1. Jarak Tempuh Maksimum

Dari *Origin* Atapupu ke *Destination* Kab. Badung:

- a. Jarak Tempuh Aktual = 1911 km
- b. Demand badung tidak perlu diperhatikan karena jarak actual ≥ 200 km
- c. Berdasarkan tabel Rasio Jarak Tempuh, beban biaya yang dikenakan sebesar @Rp 9.350,00

2. Jarak Tempuh Normal

Dari *Origin* Surabaya ke *Destiantion* Kab. Bojonegoro:

- a. Jarak Tempuh Aktual = 131 km
- b. $Liter\ Own\ Use = \frac{131 \times 2}{5,4} = 48.51$
- c. $BBM\ Own\ Use = 48,518518 \times 9300 = 451,222$
- d. $Tarif\ Vendor = 50 \times 5000 = 250000$
- e. $Total\ Jumlah = 451222 + 250000 = 701222$
- f. $Grand\ Total = 701222 + 140244 = 841466$
- g. $Labours\ Supply = \frac{841466}{5000} = 168.29$
- h. $Tarif\ PerLiter = 168.29 \times 20\% = 5664$
- i. Beban biaya yang dikenakan sebesar @Rp 5.664,00

3. Jarak Tempuh Minimum

Dari *Origin* Surabaya ke *Destination* Kab.Gresik

- a. Jarak Tempuh Aktual = 32 km
- b. $Liter\ Own\ Use = \frac{100 \times 2}{5,4} = 37,037037$

- c. $BBM\ Own\ Use = 37.037037 \times 9300 = 344,444$
- d. $Tarif\ Vendor = 50 \times 5000 = 250000$
- e. $Total\ Jumlah = 344444 + 250000 = 594444$
- f. $Grand\ Total = 594444 + 118888 = 713333$
- g. $Labours\ Supply = \frac{713333}{5000} = 142.67$
- h. $Tarif\ PerLiter = 142.67 \times 20\% = 4070$
- i. Beban biaya yang dikenakan sebesar @Rp 4.070,00

4.3 Alur Distribusi

PT. Pertamina Patra Niaga mempunyai alur distribusi tersendiri pada masing-masing Region. Khususnya yang dibahas Penulis pada penelitian ini adalah PT Pertamina Patra Niaga Region Jatim Balinus. Seusai dengan namanya, PT Pertamina Patra Niaga Region Jatim Balinus hanya beroperasi di wilayah Jawa Timur, Bali dan Nusa Tenggara.

Bunker supply yang dimiliki PT Pertamina Patra Niaga Region Jatim Balinus juga terletak pada wilayah Jawa Timur, Bali dan Nusa Tenggara. Oleh karena itu, tanggung jawab distribusinya hanya meliputi wilayah tersebut. Apabila ada *demand* yang datang dan *destination/* tujuan depotnya diluar dari region yang diterangkan sebelumnya, maka yang melakukan *demand* tersebut harus menghubungi PT Patra Niaga Region yang lain yang bertanggung jawab atas wilayah *destination/* tujuan depot yang diinginkan.

Proses distribusi dari *origin/ bunker* ke *destination/* depot tujuan ada 2 macam yaitu:

- I. *Origin/ Bunker (Supply)* → Mobil Tangki Pertamina → *Destination/ Depot (demand)*
- II. *Origin/ Bunker (Supply)* → Kapal Tanker Pertamina → Pelabuhan/ *Bunker* → Mobil Tangki Pertamina → *Destination/ Depot (demand)*

Setiap proses distribusi tersebut diberlakukan berdasarkan kondisi ratio jarak tempuh dari masing- masing distribusi seperti yang tercantum pada Tabel 4.4

Tabel 4.4 Tabel Kondisi Proses Distribusi

Proses Distribusi	Kondisi Ratio Jarak Tempuh
I	Jarak Tempuh Normal dan Jarak Tempuh Minimum
II	Jarak Tempuh Maksimum

4.4 Penyusunan Model Matematis

Dalam studi kasus ini model matematis yang digunakan seperti bab 2.2.6 yaitu sebagai berikut:

Tabel 4.5 Matriks Transportasi Model Matematis

Destination Origin	D ₁		D ₂		...	D _n		Supply b _i
O ₁	x ₁₁	C ₁₁	x ₁₂	C ₁₂	...	x _{1n}	C _{1n}	b ₁
		C ₂₁		C ₂₂			C _{2n}	
O ₂	x ₂₁	C ₂₁	x ₂₂	C ₂₂	...	x _{2n}	C _{2n}	b ₂
...
O _m	x _{m1}	C _{m1}	x _{m2}	C _{m2}	...	x _{mn}	C _{mn}	b _m
Demand								$\sum b_i$
a _j	a ₁		a ₂		...	a _n		$\sum a_j$

Notasi:

D = Tujuan/ *Destinasi* pengiriman

O = Asal/ *Origin* Pengiriman

b_i = Kapasitas *supply bunker* i

a_j = Jumlah kebutuhan *demand* di tujuan j

C_{ij} = biaya yang dibebankan dari *bunker* i ke tujuan j

X_{ij} = banyaknya bbm yang dikirimkan dari supply i ke tujuan j

Fungsi Tujuan:

Meminimalkan biaya distribusi:

$$\text{Min } Z = \sum_{i=1}^n \sum_{j=1}^m C_{ij} X_{ij}$$

Constrains:

1. Jumlah BBM yang dikirimkan harus lebih kecil atau sama dengan supply

$$\sum_{i=1}^n X_{ij} \leq b_i$$

2. Jumlah BBM yang dikirimkan harus sama dengan demand

$$\sum_{j=1}^m X_{ij} = a_j$$

Dalam studi kasus penulis ditetapkan $n = 9$ dan $m = 41$

Variabel Keputusan:

X_{ij} = Banyaknya bbm yang dikirimkan dari supply i ke tujuan j

Halaman ini sengaja dikosongkan

BAB V IMPLEMENTASI

Pada bab ini, akan dijelaskan mengenai implementasi metode yang disusun sesuai dengan langkah-langkah yang telah dijelaskan pada bab sebelumnya.

5.1 Lingkungan Uji Coba

Uji coba optimasi dengan menggunakan algoritma dan metode *Least Cost* dilanjutkan dengan *Stepping Stone* dilakukan menggunakan bantuan Netbeans dan bahasa pemrograman Java. Komputer yang digunakan adalah sebuah laptop Intel Core i3 dengan kapasitas RAM sebesar 2 GB dan kapasitas Hard disk sebesar 500GB. Perangkat lunak yang digunakan dalam pengerjaan penelitian ini adalah sistem operasi Windows 8, dengan aplikasi komputasi Netbeans IDE 8.2, dan media penyimpanan serta penginputan data Microsoft Excel dan Notepad.txt. Pada tabel 5.1 menampilkan pereangkat keras dan lunak yang dipergunakan dalam pengerjaan penelitian tugas akhir ini.

Tabel 5.1 Tabel Perangkat Keras dan Lunak yang digunakan

Perangkat Keras	
Komputer	Laptop
Processor	Intel® Core i3
RAM	2 GB
Hard disk	500 GB
Perangkat Lunak	
Sistem Operasi	Windows 8
Applikasi Java	Netbeans IDE 8.2
Media Penyimpanan Data	Microsoft Excel 2013
Media Penginputan Data	Notepad.txt

5.2 Pengimplementasian Java

Pada tahap ini pengimplementasian java ini, penjabaran dari kode program yang dibuat penulis akan ditampilkan bergantung dari kegunaan algoritma masing-masing metode.

5.3 Pembentukan Matriks Awal

Pembentukan matrix awal pada algoritma ini disertai dengan proses penginputan data yang akan didistribusikan ke kolom-kolom matrix. Dimana data yang dibutuhkan adalah jumlah sumber *supply* dan sumber *demand*, jumlah *supply* dan *demand* dari masing masing matrix, serta beban biaya dari *supply* i ke *demand* j.

5.3.1 Algoritma Matriks

Pada tahap ini akan dijabarkan algoritma yang digunakan untuk membuat matrix transportasi.

```

static void init(String filename) throws Exception {
    try {Scanner sc = new Scanner(new File(filename)); {
        int origin = sc.nextInt();
        int destination = sc.nextInt();

        List<Integer> ori = new ArrayList<>();
        List<Integer> dst = new ArrayList<>();

        for (int i = 0; i < origin; i++)
            ori.add(sc.nextInt());

        for (int i = 0; i < destination; i++)
            dst.add(sc.nextInt());

        supply = ori.stream().mapToInt(i -> i).toArray();
        demand = dst.stream().mapToInt(i -> i).toArray();

        biaya = new Double[supply.length][demand.length];
        matrix = new Transport[supply.length][demand.length];

        for (int i = 0; i < origin; i++)
            for (int j = 0; j < destination; j++)
                biaya[i][j] = sc.nextDouble();
    }
    }
}

```

Kode Program 5.1 Algoritma Matriks

Pada kode program 5.1 baris no.1 menggunakan fungsi scanner yang akan dijelaskan lebih lanjut pada metode input di bab 5.3.2 sementara hasil dari input tersebut akan membentuk besar dari matrix transport tersebut. 2 input tersebut adalah besar n dan m dari matrix pada gambar 4.5

Sedangkan pada baris ke 2 memberi nilai b_i ke m dan a_j ke n secara berurutan. Dan pada baris ke 3 menentukan C_{ij} setiap kolom dengan variabel biaya serta variabel matrix pada source code nantinya akan menentukan pengalokasian variabel keputusan.

5.3.2 Metode Input

Metode input yang digunakan sesuai dengan *source code* pada gambar 5.1 dimana string filename merupakan directory dari input yang dibutuhkan fungsi scanner, untuk menjalankannya dibutuhkan 1 file yang memuat semua kebutuhan input dalam bentuk text. Dalam hal ini penulis menggunakan file berformat txt.

5.4 Algoritma *Least Cost*

Pada tahap ini merupakan dasar algoritma yang digunakan penulis. Pembuatan algoritma *least cost* bertujuan untuk mengoptimasi biaya distribusi yang diperlukan. Dengan mengalokasikan semua supply yang bisa ke dalam kolom yang memerlukan biaya distribusi yang paling rendah yang ada untuk memenuhi demand. Dengan demikian biaya distribusi yang dibutuhkan akan lebih rendah dibanding sebelumnya. Berikut merupakan source code dari algoritma *least cost* dibuat sedemikian rupa agar prinsip metode *least cost* dapat berjalan:

```

static void LeastCostRule() {
    Double [][] temp = cloneArray(biaya);
    for (int a = 0; a < supply.length; a++)
        for (int d = 0; d < demand.length; d++) {

            int[] index = minCostIndex(temp);
            temp[index[0]][index[1]] = Double.NaN;

            int quantity = Math.min(supply[index[0]], demand[index[1]]);
            if (quantity > 0) {
                string[] index01 = new Transport(quantity, biaya[index[0]][index[1]], index[0], index[1]);

                supply[index[0]] -= quantity;
                demand[index[1]] -= quantity;
                guiTransport gui = new guiTransport();
            }
        }
}

```

Kode Program 5.2 Algoritma *Least Cost*

Dalam algoritma tersebut menggunakan clonearray dan fungsi temp agar dapat menemukan biaya terendah dalam 2 dimensional array seperti matrix yang terbentuk dan mengganti nilai terkecil yang ditemukan. Dengan begitu masih dapat menghitung solusi menggunakan array awal.

```

static Double[][] cloneArray(Double[][] src) {
    int length = src.length;
    Double[][] target = new Double[length][src[0].length];
    for (int i = 0; i < length; i++) {
        System.arraycopy(src[i], 0, target[i], 0, src[i].length);
    }
    return target;
}

```

Kode Program 5.3 Source Code Clonearray

```

static int[] minCostIndex(Double[][] array){
    Double min = array[0][0];
    int[] minIndex = {0,0};
    Double max = array[0][0];
    int[] maxIndex = {0,0};

    for (int i=0; i<array.length; i++)
    {
        for (int j=0; j<array[i].length; j++)
        {
            if (array[i][j] < min)
            {
                min = array[i][j];
                minIndex[0] = i;
                minIndex[1] = j;
            }
            if (array[i][j] > max) {
                max = array[i][j];
                maxIndex[0] = i;
                maxIndex[1] = j;
            }
        }
    }

    return minIndex;
}

```

Kode Program 5.4 MinCostIndex pengganti nilai terkecil

Pada source code mincostindex seperti kode program 5.4 digunakan untuk mencari nilai terkecil dan menggantinya dengan maxvalue temporary pada source code 5.2

5.5 Algoritma *Stepping Stone*

Tahap selanjutnya mengimplementasikan algoritma *Stepping Stone*. Penerapan algoritma *stepping stone* ini bertujuan untuk lebih mengoptimalkan alokasi yang telah diterapkan pada metode *least cost* sebelumnya. Dengan melihat “*opportuniy cost*” seperti yang sudah dijelaskan pada metodologi, perpindahan alokasi dari kolom satu ke kolom yang lain akan mengakibatkan perubahan biaya. Dan perubahan biaya pada “*opptrtunity cost*” sendiri dapat membuat biaya distribusi semakin kecil sehingga hasilnya akan lebih optimal. Kode program dari *stepping stone* seperti gambar-gambar dibawah ini:

```

static void steppingStone() {
    double maxReduction = 0;
    Transport[] move = null;
    Transport leaving = null;

    fixDegenerateCase();

    for (int s = 0; s < supply.length; s++) {
        for (int d = 0; d < demand.length; d++) {

            if (matrix[s][d] != null)
                continue;

            Transport trial = new Transport(0, biaya[s][d], s, d);
            Transport[] path = getClosedPath(trial);

            double reduction = 0;
            double lowestQuantity = Integer.MAX_VALUE;
            Transport leavingCandidate = null;

            boolean plus = true;
            for (Transport x : path) {
                if (plus) {
                    reduction += x.biayaPerLiter;
                } else {
                    reduction -= x.biayaPerLiter;
                    if (x.quantity < lowestQuantity) {

```

Kode Program 5.5 Algoritma *Stepping Stone* bag.1

```

                plus = !plus;
            }
            if (reduction < maxReduction) {
                move = path;
                leaving = leavingCandidate;
                maxReduction = reduction;
            }
        }
    }

    if (move != null) {
        double q = leaving.quantity;
        boolean plus = true;
        for (Transport x : move) {
            x.quantity += plus ? q : -q;
            matrix[x.s][x.d] = x.quantity == 0 ? null : x;
            plus = !plus;
        }
        steppingStone();
    }
}

```

Kode Program 5.6 Source Code Algoritma *Stepping Stone* bag.2

Pada kode program 5.5 dan 5.6 menunjukkan algoritma *stepping stone* dijalankan. Menggunakan *fixdegeneratecase* yang akan dibahas selanjutnya. Dalam algoritma tersebut agar matrix yang sudah terbentuk pada *least cost* untuk di optimasi lagi menggunakan algoritma *stepping stone*. Kemudian Algoritma tersebut diulang-ulang sedemikian rupa sampai *fixdegeneratecase* ter *trigger*.

5.5.1 Fix Degenerate Case

Tahap ini yang memastikan algoritma *stepping stone* untuk melihat apabila *feasible solution* lebih sedikit dibanding $m+n-1$ nilai positif di X_{ij} .

```
static void fixDegenerateCase() {
    final double eps = Double.MIN_VALUE;

    if (supply.length + demand.length - 1 != matrixToList().size()) {

        for (int s = 0; s < supply.length; s++)
            for (int d = 0; d < demand.length; d++) {
                if (matrix[s][d] == null) {
                    Transport dummy = new Transport(eps, biaya[s][d], s, d);
                    if (getClosedPath(dummy).length == 0) {
                        matrix[s][d] = dummy;
                        return;
                    }
                }
            }
    }
}
```

Kode Program 5.7 Fix Degenerate Case

5.5.2 Penentuan Sel Matriks *Stepping Stone*

Tahap selanjutnya menentukan sel matrix mana saja yang dapat dilalui *stepping stone* dan penempatan nilai plus dan minus seandainya menemukan sel matrix yang sesuai.

```

static LinkedList<Transport> matrixToList() {
    return stream(matrix)
        .flatMap(row -> stream(row))
        .filter(x -> x != null)
        .collect(toCollection(LinkedList::new));
}

static Transport[] getClosedPath(Transport x) {
    LinkedList<Transport> path = matrixToList();
    path.addFirst(x);

    // remove (and keep removing) elements that do not have a
    // vertical AND horizontal neighbor
    while (path.removeIf(e -> {
        Transport[] nbrs = getNeighbors(e, path);
        return nbrs[0] == null || nbrs[1] == null;
    }));

    // place the remaining elements in the correct plus-minus order
    Transport[] stones = path.toArray(new Transport[path.size()]);
    Transport prev = x;
    for (int i = 0; i < stones.length; i++) {
        stones[i] = prev;
        prev = getNeighbors(prev, path)[i % 2];
    }
    return stones;
}

```

Kode Program 5.8 Penentuan Sel Matriks Stepping stone

5.6 Pengalokasian Demand pada Matriks

Dalam tahapan ini, seperti yang telah dijabarkan pada gambar 5.1 baris ke 3, variabel *matrix* pada source code akan menempatkan pengalokasian *hsd* pada kolom-kolom yang telah terbuat pada *matrix* awal secara tepat. Untuk lebih jelasnya pada masing-masing algoritma *least cost* dan *stepping stone* dapat dilihat seperti gambar berikut.

```

static void leastCostRule() {
    Double [][] temp = cloneArray(hsdp);
    for (int s = 0; s < supply.length; s++) {
        for (int d = 0; d < demand.length; d++) {

            int[] index = minCostIndex(temp);
            temp[index[0]][index[1]] = Double.MAX_VALUE;

            int quantity = Math.min(supply[index[0]], demand[index[1]]);
            if (quantity > 0) {
                xsdp[index[0]][index[1]] = new xsdp(quantity, hsdp[index[0]][index[1]], index[0], index[1]);

                supply[index[0]] -= quantity;
                demand[index[1]] -= quantity;
                gxiTransport gxi = new gxiTransport();
            }
        }
    }
}

```

Kode Program 5.9 Pengalokasian Demand pada *Least Cost*

Jadi pengalokasian hsd pada masing-masing kolom bernama quantity, dimana nantinya akan dikurangi supply dan demand sesuai kolom X_{ij} agar jumlah alokasinya tepat dengan kebutuhan dimana tidak melebihi supply, dan mengurangi demand sesuai kolom.

Sedangkan pengalokasian pada *stepping stone* dapat dilihat pada gambar 5.5 untuk perpindahan kuota hsd ke kolom yang dituju, dan pada gambar 5.6, kolom yang telah ditinggalkan akan dikurangi dengan jumlah kuota yang telah berpindah ke kolom yang dituju.

5.7 Algoritma Solusi Tujuan

Tahanapan ini berisikan source code penghitungan total cost dari matrix yang terbentuk. Dimana angka yang masuk pada matrix X_{ij} di kalikan dengan C_{ij} kemudian dijumlahkan.

```

static void printResult(String filename) {

    double totalCosts = 0;

    for (int s = 0; s < supply.length; s++) {
        for (int d = 0; d < demand.length; d++) {

            Transport x = matrix[s][d];
            if (x != null && x.s == s && x.d == d) {
                totalCosts += (x.quantity * x.biayaPerLiter);
            } else
                System.out.printf(" - ");

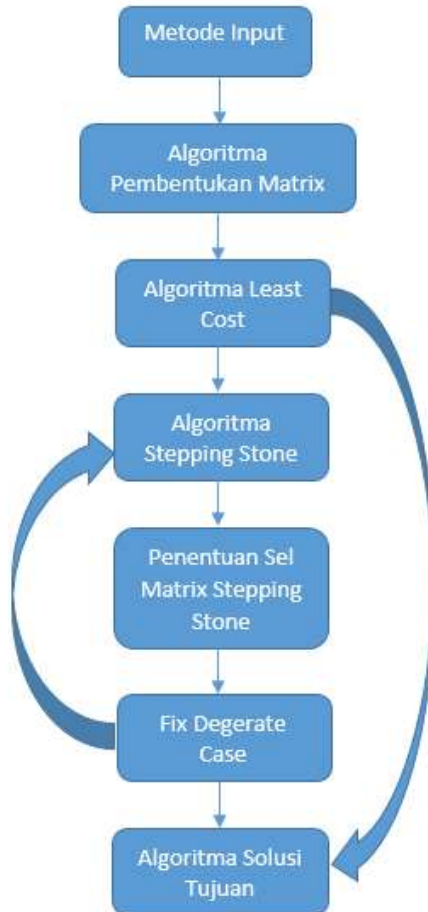
        }
        System.out.println();
    }
    System.out.printf("%nTotal biaya: %t%n", totalCosts);
}

```

Kode Program 5.10 Penghitungan Total Cost/Solusi

5.8 Flow Pengolahan Data

Pada tahap ini akan dijelaskan flow dari data masukan sehingga dapat menghasilkan solusi untuk menyelesaikan masalah transportasi. Flow dijelaskan berdasarkan tahapan-tahapan pengkodean yang sudah dijelaskan sebelumnya. Dan dapat dilihat pada gambar 5.1.



Gambar 5.1 Source Code Flow Olah Data

5.8.1 Flow Metode Input

Flow awal dari data masukkan dimulai dari metode input, pada tahap ini seperti kode program 5.1 data bisa masuk menggunakan fungsi scanner pada java. Inputan yang di

masukkan berupa file.txt yang akan dijelaskan format filenya pada bagian *User Interface* selanjutnya.

5.8.2 Flow Algoritma Matriks

Pada tahap ini sebenarnya dilakukan secara bersamaan pada metode input, karena data yang diterima dari scanner langsung diarahkan ke sel-sel matrix secara berurutan. Sehingga besar matrix sesuai dengan data masukkan.

5.8.3 Flow Algoritma *Least Cost*

Dengan masukan pada variabel biaya sesuai matrix yang telah terbentuk berdasarkan algoritma matrix, akan dialokasikan hsd sesuai algoritma *least cost* yang telah dibahas sebelumnya. Jadi dapat terlihat variabel keputusannya pada solusi awal

.

5.8.4 Flow Algoritma *Stepping Stone*

Setelah terbentuk solusi awal dan matrix pada algoritma *least cost*. Variabel keputusan akan diolah lagi sesuai dengan algoritma *stepping stone*. Didalam algoritma *stepping stone* juga terdapat algoritma penentuan sel *stepping stone* dan fix degenerate case. Jadi pada tahap ini sebenarnya dilakukan looping dari ke 3 algoritma tersebut.

5.8.5 Flow Penentuan Sel *Stepping Stone*

Pada tahap ini akan ditentukan sel mana yang jumlah alokasi variabel keputusannya harus dipindah. Dimulai dari melihat sel yang kosong dan menilai *opportunity cost* dari masing-masing sel kosong tersebut. Dimana nilai negatif terbesar yang menjadi pilihan untuk perpindahan alokasi variabel keputusan. Apabila tidak ditemukan nilai negatif pada *opportunity cost* maka looping berhenti.

5.8.6 Flow Fix Degenerate Case

Pada tahap ini, memastikan algoritma *stepping stone* untuk melihat apabila *feasible solution* lebih sedikit dibanding $m+n-1$ nilai positif di X_{ij} .

5.8.7 Flow Algoritma Solusi Tujuan

Dalam tahap ini akan dihitung setiap nilai pada X_{ij} akan dikalikan dengan biaya C_{ij} dan dijumlahkan. Jadi dapat ditemukan total biaya dari masing-masing metode. Dan juga flow ini dilakukan 2 kali, yaitu setelah flow algoritma *least cost* dan setelah flow algoritma *stepping stone*, yang bertujuan membandingkan biaya dari keduanya.

5.9 User Interface (UI) Java

Pada tahap ini penulis menggunakan fitur JFrame Form pada aplikasi Net Beans untuk membuat UI (*user interface*) matrix *least cost* dan *stepping stone*. Dengan menggunakan UI dapat mempermudah proses input data dan menampilkan matrix metode-metode yang dipakai penulis secara lebih informatif.

Berikut merupakan tampilan UI keseluruhan dari matrix *least cost* dan *stepping stone* pada gambar 5.2:

Gambar 5.2 UI *Least Cost & Stepping Stone*

5.10 Fungsi-Fungsi pada UI *Least Cost & Stepping Stone*

Seperti yang terlihat pada gambar 5.2, tampilan UI tersebut pada dasarnya dibuat untuk menyelesaikan permasalahan yang diterangkan penulis pada penelitian ini, tetapi ada juga berbagai macam fungsi yang diterapkan agar UI tersebut lebih fleksibel apabila user ingin memberikan input/masukkan yang berbeda demi membantu melakukan optimasi dengan metode transportasi sesuai kepentingan perusahaan tersebut.

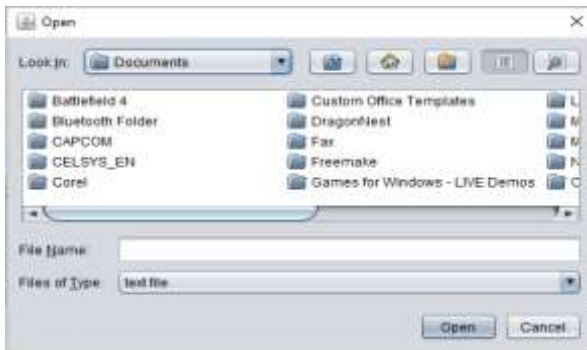
Selanjutnya data yang dimasukkan nantinya akan diolah dengan algoritma yang sudah dijelaskan sebelumnya dan pada UI tersebut akan ditampilkan keluaran berupa matrix-matrix yang terisi pada tabel pada UI. Dengan demikian informasi keluaran optimasi dengan metode-metode yang digunakan penulis dapat tersampaikan dengan sebaik-baiknya.

5.10.1 Fungsi Pembentukan Matriks dan File Masukkan

Pada tahap ini disediakan kolom untuk user agar dapat menentukan file masukkan yang nantinya akan menentukan besar matrix transportasi sesuai kebutuhannya.

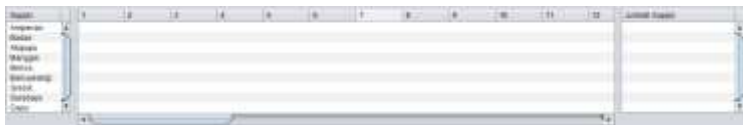
Gambar 5.3 UI Choose File dan Download Contoh Format

Pada bagian ini user dapat menentukan input yang akan diolah dalam matrix transportasi. Masukkan yang digunakan berupa file.txt dan disertakan contoh format masukkan sesuai kebutuhan algoritma sebelumnya. Seperti gambar 5.3 apabila user menekan tombol *Choose File* nantinya akan diteruskan dengan file chooser yang mempermudah user dalam memilih file.txt untuk masukkan yang akan digunakan.



Gambar 5.4 UI File Chooser

Sesuai dengan inputan yang dipilih pada gambar 5.4, sistem akan membuat besar matrix yang sesuai untuk menjalankan metode transportasi *least cost* dan *stepping stone*.



Gambar 5.5 UI Matriks Transportasi

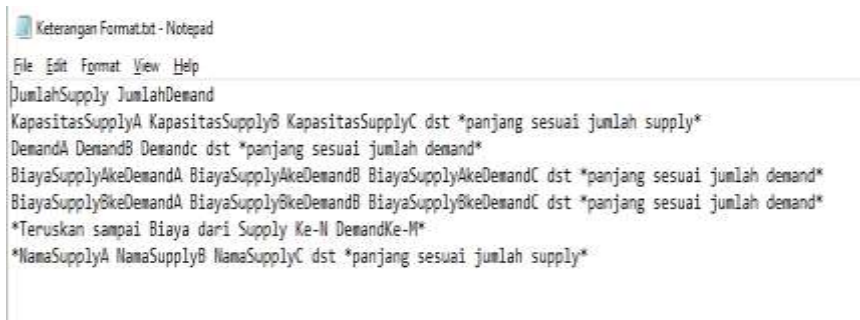
Gambar 5.5 adalah contoh matrix transportasi berukuran 9x41 sesuai dengan studi kasus yang dibahas penulis dimana ada scroll bar horizontal untuk menunjukkan kolom sampai ke kolom ke 41.

Baris *title header* 1-41 pada gambar 5.5 nantinya akan menunjukkan jumlah demand yang diinginkan pada setiap

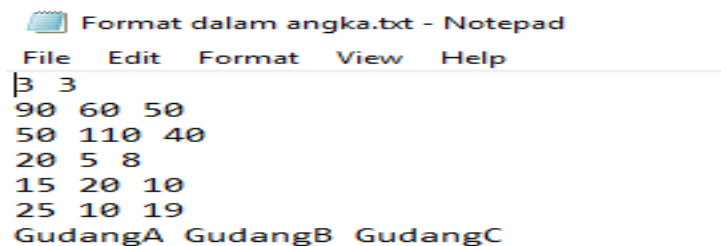
kolom berdasarkan masukan. Dan pada kolom jumlah supply juga nantinya akan terisi berdasarkan masukan user.

5.10.2 Fungsi Download Contoh Format

Pada gambar 5.3 apabila user menekan tombol download contoh format maka sistem akan membuat file contoh format file masukan yang dibutuhkan pada *directory* D:/Contoh Input/. Ada 2 contoh file masukan yaitu keterangan penulisan format yang bernama Keterangan Format.txt dan yang menunjukkan format penulisan yang sudah jadi dengan nama file Format dalam Angka.txt



Gambar 5.6 Keterangan Format.txt



Gambar 5.7 Format dalam angka.txt

5.10.3 Fungsi Matriks *Least Cost*

Pada tahap ini merupakan langkah pembentukan matrix transportasi *least cost* untuk menentukan solusi feasible awal.



Gambar 5.8 UI Matriks *Least Cost*

Pada gambar 5.8 apabila user menekan tombol Calculate maka algoritma *least cost* akan berjalan dan kemudian sistem akan menampilkan kolom-kolom yang terisi pada matrix bergantung file masukkan dan pada total biaya akan menunjukkan hasil solusi feasible awal.

5.10.4 Fungsi Matriks *Stepping Stone*

Pada tahap ini merupakan langkah pembentukan matrix transportasi *stepping stone* untuk mengoptimalkan solusi feasible awal.



Gambar 5.9 UI Matriks *Stepping Stone*, Reset Table dan Efisiensi biaya

Pada gambar 5.9 apabila user menekan tombol Calculate maka algoritma *stepping stone* akan berjalan dan kemudian sistem akan menampilkan kolom-kolom yang terisi pada matrix bergantung perpindahan sel pada algoritma *stepping stone*. Pada total biaya akan menunjukkan solusi penghitungan dari metode *stepping stone* dan pada efisiensi biaya akan

ditampilkan pengurangan biaya dari metode *least cost* ke metode *stepping stone*.

5.10.5 Fungsi Reset Table

Pada tahap ini apabila user menekan tombol Reset Table seperti gambar 5.10 akan terjadi pengosongan pada setiap matrix dan kolom masukkan user.



Gambar 5.10 UI Matriks *Stepping Stone*, Reset Table dan Efisiensi biaya

5.11 Validasi Java dan Unit Testing

Tahap ini dilakukan untuk mengetahui apakah program java yang dibuat telah berjalan sebagaimana mestinya dan algoritma yang telah dibuat benar-benar bisa berjalan sesuai fungsi yang telah dibuat.

Dengan membandingkan pengerjaan manual menurut sumber yang terpercaya dengan program yang telah dibuat. [16]

5.11.1 Perhitungan *Least Cost*

Berikut matriks awal dari masalah yang ingin dibandingkan:

Tabel 5.1 Matriks transportasi validasi

To From	A	B	C	Supply
1	6	8	10	150
2	7	11	11	175
3	4	5	12	275
Demand	200	100	300	600

Kemudian langkah berikutnya yaitu melihat hasil dari matriks diatas dengan menggunakan metode *Least Cost*.

Tabel 5.2 Matriks transportasi *least cost* validasi

To From	A	B	C	Supply
1	6	8	10	150
2	7	11	11	175
3	4	5	12	275
Demand	200	100	300	600

Setelah terlihat hasil seperti diatas, dilanjutkan dengan membandingkan apakah program yang dibuat dapat menyelesaikan dengan hasil yang sama.

Sebelum menjalankan program yang dibuat sesuai dengan permasalahan diatas, maka harus dibuat masukkan pada

notepad.txt agar dapat dijalankan pada program seperti gambar 5.11.

```
3 3
150 175 275
200 100 300
6 8 10
7 11 11
4 5 12|
GudangA GudangB GudangC
```

Gambar 5.11 Masukkan Validasi Java

Setelah dibuat masukkan dari notepad.txt tersebut barulah bisa dijalankan untuk melihat hasil *least cost* pada java.

Supply	1d:200	2d:100	3d:300
GudangA		25.0	125.0
GudangB			175.0
GudangC	200.0	75.0	

Gambar 5.12 Hasil penghitungan *least cost*

Pada tahap ini dapat diketahui untuk fungsi algoritma *least cost* pada java telah berjalan sebagaimana mestinya dilihat dari hasil solusi awal yang sama.

5.11.2 Perhitungan *Stepping Stone*

Kemudian langkah selanjutnya adalah membandingkan hasil *stepping-stone* dari sumber tersebut dan program java yang telah dibuat.

Tabel 5.3 Matriks transportasi *stepping stone* validasi

To From	A	B	C	Supply
1	+ 6 25	8	- 10 125	150
2	7	11	11 175	175
3	- 4 175	5 100	+ 12	275
Demand	200	100	300	600

Langkah selanjutnya yaitu dengan melakukan metode tersebut menggunakan algoritma *stepping stone* pada java yang telah dibuat.

Supply	1d:200	2d:100	3d:300
GudangA	25.0		125.0
GudangB			175.0
GudangC	175.0	100.0	

Gambar 5.13 Hasil penghitungan *stepping stone*

Setelah metode *stepping stone* dilakukan pada java dapat terlihat bahwa penghitungan manual pada sumber tersebut dan penghitungan pada java menunjukkan hasil yang sama, dengan demikian maka program yang telah dibuat adalah valid.

5.11.3 Java *Stepping Stone Loop Output*

Untuk mengetahui apakah hasil dari *stepping stone* dari java sudah benar-benar final dan tidak ada hasil yang lebih kecil lagi, penulis menjabarkan 3 hasil looping terakhir dari java pada metode *stepping stone* yang dijalankan.

Output - Least Cost & Ste	Output - Least Cost & Ste	Output - Least Cost & Stepp
3.2422005E9	3.8352005E9	1.9376005E9
3.7771055E9	3.2422005E9	2.8352005E9
6.1857455E9	3.7771055E9	3.2422005E9
6.2264455E9	6.1857455E9	3.7771055E9
6.8213981E9	6.2264455E9	6.1857455E9
6.8411376E9	6.8213981E9	6.2264455E9
6.8859076E9	6.8411376E9	6.8213981E9
9.4824455E9	6.8859076E9	6.8411376E9
1.27384455E10	9.4824455E9	6.8859076E9
1.34832555E10	1.27384455E10	9.4824455E9
1.36685855E10	1.34832555E10	1.27384455E10
1.38517355E10	1.36685855E10	1.34832555E10
1.39697655E10	1.38517355E10	1.36685855E10
1.42028585E10	1.39697655E10	1.38517355E10
1.90868585E10	1.42028585E10	1.39697655E10
1.91926785E10	1.90868585E10	1.42028585E10
2.02264585E10	1.91926785E10	1.90868585E10
2.07189285E10	2.01654085E10	1.91926785E10
2.28923085E10	2.06578785E10	2.01043585E10
2.30713885E10	2.28312585E10	2.05968285E10
2.35475785E10	2.30103385E10	2.27702085E10
2.51918585E10	2.34865285E10	2.29492885E10
2.56654905E10	2.51918585E10	2.34254785E10
2.56858405E10	2.56654905E10	2.51918585E10
3.25939365E10	2.56858405E10	2.56654905E10
3.30721605E10	3.25939365E10	2.56858405E10
3.31639305E10	3.30721605E10	3.25939365E10
3.32652915E10	3.31639305E10	3.30721605E10
3.37414815E10	3.32652915E10	3.31735215E10
3.623184455E10	3.37414815E10	3.36497115E10
3.625290956E10	3.632274456E10	3.632187456E10
3.637093956E10	3.634380956E10	3.634293956E10
3.639200456E10	3.646183956E10	3.646096956E10
3.642456456E10	3.648290456E10	3.648203456E10
3.707169456E10	3.651546456E10	3.651459456E10
3.716608956E10	3.716259456E10	3.716172456E10
3.729632956E10	3.729283456E10	3.729196456E10
3.771446956E10	3.771097456E10	3.771010456E10
4.258625956E10	4.258276456E10	4.258189456E10

Gambar 5.14 Hasil output java stepping stone loop

Pada gambar 5.14 adalah 3 hasil looping terajhir pada *stepping stone*. Disetiap looping pada java tersebut dilakukan untuk menemukan biaya yang lebih kecil dari hasil looping sebelumnya dan akan berhenti apabila tidak ada lagi hasil yang lebih kecil proses *looping* akan berhenti. Karena pada dasarnya metode *stepping stone* akan mencari biaya yang lebih rendah pada setiap prosesnya, apabila pada proses selanjutnya biaya yang didapatkan lebih besar, maka proses tersebut dihentikan dan bisa disebut sebagai hasil yang paling optimal.

Halaman ini sengaja dikosongkan

BAB VI

HASIL DAN PEMBAHASAN

Pada bab ini, akan dijelaskan mengenai hasil yang didapatkan dari penelitian ini dan pembahasan terhadap hasil tersebut secara keseluruhan.

6.1 Masukkan Metode *Least Cost & Stepping Stone*

Berikut merupakan masukkan untuk diolah pada matrix transportasi *least cost & stepping stone* berupa jumlah supplier, jumlah demand, jumlah masing masing demand, dan beban biaya masing masing kolom matrix yang didapat pada penghitungan beban biaya. Penulisan pada file.txt harus sesuai format yang dijelaskan sebelumnya agar dapat diolah dengan algoritma secara tepat.

File masukkan dipilih melalui file chooser secara berurutan seperti pada gambar 5.3 kemudian gambar 5.4

6.2 Pembentukan Matriks *Least Cost*

Tampilan UI awal matrix *least cost* setelah jumlah supplier, *jumlah demand dan file masukkan ditentukan



Gambar 6.3 UI Matriks awal *least cost*

Seperti yang terlihat pada gambar 6.3 matrix awal *least cost* telah terbentuk. Pada baris *title header* 1-41 sudah tertera demand yang diinginkan pada masing masing kolom. Dan pada kolom jumlah supply sudah tertera supply yang dimiliki masing masing *bunker*. Baris dan kolom tersebut secara otomatis akan terupdate apabila file inputan telah dipilih dan kolom supplier dan demand terisikan dengan benar.

Selanjutnya dengan menekan tombol calculate, secara otomatis akan terisi banyaknya bbm yang dikirimkan ke masing masing tujuan dari supply dan penghitungan solusi feasible awal dari matrix transportasi *least cost* seperti gambar berikut:

Supply	1d148180	2d82000	3d183000	4d5000	5d4850	6d11960	7d43000	8d45000	9d82000	10d15000	11d23000	12d50	Jumlah Supply
Atapupu													400000
Badas													400000
Ampenan													400000
Manggis	140180.0				4850.0								800000
Banoa													800000
Banyuwangi			183000.0				43000.0	45000.0				5000.0	1500000
Gresik		82000.0				1118000.0							1200000
Surabaya						78000.0			82000.0		23000.0		1500000
Cepu				5000.0						15000.0			55000000
													Total Biaya:
													Rp 44524028699

Gambar 6.4 UI Matriks *Least Cost* Bagian 1

Supply	12d5000	13d1170	14d26000	15d6640	16d5000	17d29000	18d10000	19d5000	20d1210	21d5000	22d44000	23d8000	Jumlah Supply
Atapupu								5000.0					400000
Badas													400000
Ampenan													400000
Manggis													800000
Banoa													800000
Banyuwangi	5000.0										44000.0		1500000
Gresik													1200000
Surabaya			26000.0	664951.0									1500000
Cepu	117000.0				5000.0	29000.0	10000.0		121000.0	5000.0		8000.0	55000000
													Total Biaya:
													Rp 44524028699

Gambar 6.5 UI Matriks *Least Cost* Bagian 2

Supply	23d.8000	24d.5340	25d.29000	26d.1000	27d.85000	28d.60000	29d.5000	30d.4000	31d.1590	32d.21000	33d.1911	34d.11	Jumlah Supply
Atapupu							5000.0				132230.0		400000
Badas			100000.0		85000.0								400000
Ampenan											300000.0		400000
Manggis						60000.0					588970.0		800000
Benoa											800000.0		800000
Banjuwangi			29000.0										1500000
Gresik													1200000
Surabaya		505049.0											1500000
Cepu	8000.0	28951.0						4000.0	159000.0	21000.0		15000	55000000
													Total Biaya: Rp 44524028699

Gambar 6.6 UI Matriks *Least Cost* Bagian 3

Supply	3000	31d.1590	32d.21000	33d.1911	34d.15000	35d.32000	36d.44000	37d.10000	38d.59000	39d.1170	40d.1197	41d.1730	Jumlah Supply
Atapupu				132230.0			44000.0				117000.0		400000
Badas													400000
Ampenan				300000.0				10000.0					400000
Manggis				588970.0									800000
Benoa				800000.0									800000
Banjuwangi													1500000
Gresik													1200000
Surabaya													1500000
Cepu	0	159000.0	21000.0		15000.0	32000.0			59000.0		1197000.0	1730000.0	55000000
													Total Biaya: Rp 44524028699

Gambar 6.7 UI Matriks *Least Cost* Bagian 4

6.3 Pembentukan Matriks *Stepping Stone*

Setelah matrix *least cost* terbentuk dan solusi feasible awal ditemukan, langkah selanjutnya adalah menentukan solusi optimasi dengan menggunakan metode *stepping stone* dengan menekan tombol calculate pada matrix *stepping stone*.

Setelah menekan tombol calculate pada matrix *stepping stone* akan ditemukan solusi akhir yang telah dioptimasi menggunakan metode *stepping stone* dan akan terisi banyaknya bbm yang dikirimkan ke masing masing tujuan dari supply berdasarkan matrix *least cost* sebelumnya kemudian dipindahkan ke sel lain sesuai algoritma *stepping stone*. Setelah itu secara otomatis akan terhitung solusi akhir pada total biaya dan pada efisiensi biaya juga akan terisi berdasarkan perbandingan biaya dari *least cost* dan *stepping stone*.

Berikut merupakan gambar matrix *stepping stone* beserta solusi akhir dan efisiensi biaya:

Supply	1d:146180	2d:82000	3d:183000	4d:5000	5d:4850	6d:11950	7d:43000	8d:45000	9d:82000	10d:15000	11d:23000	12d:5000	Jumlah Supply
Atapupu										15000.0		5000	400000
Badas													400000
Ampenan													400000
Manggis	146180.0				4850.0								800000
Benoa													800000
Banyuwangi			183000.0				43000.0	45000.0					1500000
Gresik													1200000
Surabaya													1500000
Cepu		82000.0		5000.0		1195000.0			82000.0		23000.0		5500000
<div>◀──</div>													

Supply	23d.8000	24d.5340...	25d.29000	26d.1000...	27d.85000	28d.09000	29d.5000	30d.4000	31d.1580...	32d.21000	33d.1911...	34d.1	Jumlah Supply
Atapupu					4 9F-324		5000 0	4000 0		21000 0	83230 0	15000 0	400000
Badas			100000 0		85000 0								400000
Ampenan											300000 0		400000
Manggis						11000 0					637970 0		800000
Benda											800000 0		800000
Banyuwangi			29000 0			49000 0							1500000
Gresik													1200000
Surabaya		534000 0											1500000
Cepu	8000 0								159000 0				55000000
<div> <div>RESET TABLE</div> <div> <div>Total Biaya:</div> <div>Rp 42581894560</div> </div> <div> <div>Efisiensi Biaya:</div> <div>Rp 1942134139</div> </div> </div>													

Gambar 6.10 UI Matriks *Stepping Stone* Bagian 3

Supply	300	31d.1580...	32d.21000	33d.1911...	34d.15000	35d.32000	36d.44000	37d.10000	38d.89000	39d.1170...	40d.1187...	41d.1730...	Jumlah Supply
Atapupu	0		21000 0	83230 0	15000 0							96000 0	400000
Badas													400000
Ampenan				390000 0			10000 0						400000
Manggis				637970 0									800000
Benda				800000 0									800000
Banyuwangi													1500000
Gresik												1200000 0	1200000
Surabaya							44000 0			117000 0		434000 0	1500000
Cepu	159000 0				32000 0			69000 0			1187000 0		55000000
<div> <div>RESET TABLE</div> <div> <div>Total Biaya:</div> <div>Rp 42581894560</div> </div> <div> <div>Efisiensi Biaya:</div> <div>Rp 1942134139</div> </div> </div>													

Gambar 6.11 UI Matriks *Stepping Stone* Bagian 4

Hasil dari program diatas menunjukkan bahwa total biaya pada *stepping stone* lebih rendah daripada *least cost*, dimana perbedaan biaya tersebut terlihat pada kolom efisiensi biaya. Jadi penggunaan metode *stepping stone* sebagai metode optimasi setelah *least cost* dapat lebih meminimalkan biaya.

6.4 Hasil Alokasi Metode *Least Cost*

Setelah menjalankan program seperti pada gambar 6.4, 6.5, 6.6 dan 6.7. Dapat dilihat hasil dari alokasi masing-masing *origin* ke *destination* yang ada. Kemudian dari gambar tersebut dapat dicantumkan pada tabel alokasi seperti pada tabel 6.1 dibawah:

Tabel 6.1 Alokasi *least cost*

Origin	Alokasi	
	Destinasi	Jumlah
Atapupu	19(Pamekasan)	5000
	29(Temanggung)	5000
	33(Denpasar)	132230
	36(Kodya Malang)	44000
	39(Pasuruan)	117000
Badas	26(Sumbawa)	100000
	27(Sumbawa Barat)	85000
Ampenan	33(Denpasar)	390000
	37(Mataram)	10000
Manggis	1(Badung)	146180
	5(Gianyar)	4850
	28(Tabanan)	60000
	33(Denpasar)	588970
Benoa	33(Denpasar)	800000
Banyuwangi	3(Banyuwangi)	183000
	7(Jember)	43000
	8(Jembrana)	45000
	12(Lumajang)	5000

	22(Probolinggo)	44000
	25(Situbondo)	29000
Gresik	2(Bangkalan)	82000
	6(Gresik)	1118000
Surabaya	6(Gresik)	78000
	9(Jombang)	82000
	11(Lamongan)	23000
	14(Kab.Malang)	26000
	15(Kab.Mojokerto)	664951
	20(Pasuruan)	121000
	24(Kab.Sidoarjo)	505049
Cepu	4(Bojonegoro)	5000
	10(Kab.Kediri)	15000
	13(Kab.Madiun)	117000
	16(Nganjuk)	5000
	17(Ngawi)	29000
	18(Pacitan)	10000
	21(Ponorogo)	5000
	23(Rebang)	8000
	24(Kab.Sidoarjo)	28951
	30(Trenggalek)	4000
	31(Tuban)	159000
	32(Tulungagung)	21000
	34(Kodya Kediri)	15000
	35(Kodya Madiun)	32000
	38(Kodya Mojokerto)	69000
	40(Kodya Probolinggo)	1197000
	41(Kodya Surabaya)	1730000

6.5 Hasil Alokasi Metode *Stepping Stone*

Kemudian selanjutnya untuk mengetahui alokasi dari metode *stepping stone* dapat dilihat pada gambar 6.8, 6.9, 6.10 dan 6.11. dan dijabarkan pada tabel 6.2:

Tabel 6.2 Alokasi *Stepping Stone*

Origin	Alokasi	
	Destinasi	Jumlah
Atapupu	10(Kab.Kediri)	15000
	12(Lumajang)	5000
	18(Pacitan)	10000
	19(Pamekasan)	5000
	22(Probolinggo)	44000
	29(Temanggung)	5000
	30(Trenggalek)	4000
	32(Tulungagung)	21000
	33(Denpasar)	83230
	34(Kodya Kediri)	15000
	41(Kodya Surabaya)	95000
Badas	26(Sumbawa)	100000
	27(Sumbawa Barat)	85000
Ampenan	33(Denpasar)	390000
	37(Mataram)	10000
Manggis	1(Badung)	146180
	5(Gianyar)	4850
	28(Tabanan)	11000
	33(Denpasar)	637970
Benoa	33(Denpasar)	800000
Gresik	41(Kodya Surabaya)	1200000
Surabaya	14(Kab.Malang)	26000
	15(Kab.Mojokerto)	224000

	20(Pasuruan)	121000
	24(Kab.Sidoarjo)	534000
	36(Kodya Malang)	44000
	39(Pasuruan)	117000
	41(Kodya Surabaya)	434000
Cepu	2(Bangkalan)	82000
	4(Bojonegoro)	5000
	6(Gresik)	1195000
	9(Jombang)	82999
	11(Lamongan)	23000
	13(Kab.Madiun)	117000
	15(Kab.Mojokerto)	440951
	16(Nganjuk)	5000
	17(Ngawi)	29000
	21(Ponorogo)	5000
	23(Rembang)	8000
	31(Tuban)	1159000
	35(Madiun)	32000
	38(Kodya Mojokerto)	69000
	40(Kodya Probolinggo)	1197000

6.6 Perbandingan Biaya Eksisting dengan Biaya Hasil Metode Optimasi Transportasi

Setelah mendapatkan total biaya yang dilakukan dengan metode transportasi *least cost* dan *stepping stone*, langkah selanjutnya adalah membandingkan dengan biaya eksisting sesuai data yang didapatkan pada periode Desember 2016. Biaya yang digunakan untuk dibandingkan dengan biaya eksisting adalah biaya yang didapat dari metode *stepping stone*, karena metode tersebut lebih optimal dari *least cost*.

Berikut merupakan biaya eksisting distribusi HSD PT. Pertamina Patra Niaga Region Jatim Balinus pada Desember 2016 disertai dengan PPN dan PBBKB:

4	Row Labels	= Sum of QTY LTR	Sum of Total (IDR)	Sum of PBBKB	Sum of PPN
5	Kab. Badung	146180	876141000	7415711	78975135
6	Kab. Bangkalan	82000	485267000	4015318	43750115
7	Kab. Banyuwangi	183000	1096225000	15247971	98270691
8	Kab. Bojonegoro	5000	27816000	231415	2507713
9	Kab. Gianyar	4850	28320000	242018	2552598
10	Kab. Gresik	1196000	6852910000	54528925	618032745
11	Kab. Jember	43000	278001000	2223596	25070814
12	Kab. Jemberana	45000	265070000	2314147	33887011
13	Kab. Jombang	82000	486127000	3933666	43835810
14	Kab. Kediri	15000	89774000	743004	8093688
15	Kab. Lamongan	23000	137943000	1135403	12437082
16	Kab. Lumajang	5000	28340000	234124	2555071
17	Kab. Madiun	117000	732575000	5927401	66058378
18	Kab. Malang	26000	147342000	1195620	13285916
19	Kab. Mojokerto	664951	3774827000	29383218	340494822
20	Kab. Nganjuk	5000	31008000	247668	2796370
21	Kab. Ngawi	29000	172125000	1411940	15519403
22	Kab. Pacitan	10000	65516000	522425	5908460
23	Kab. Pamekasan	5000	30962000	1429348	2684783
24	Kab. Pasuruan	121000	710875000	5768114	64100473
25	Kab. Ponorogo	5000	31008000	247668	2796370
26	Kab. Probolinggo	44000	259979000	2150843	23438995
27	Kab. Rembang	8000	49098000	398127	4427286

Gambar 6.12 Biaya Eksisting Bagian 1

28	Kab. Sidoarjo	534000	3234468000	26003711	291678358
29	Kab. Situbondo	29000	161315000	1322084	14544733
30	Kab. Sumbawa	100000	604250000	4676668	54506675
31	Kab. Sumbawa Barat	85000	510547000	3951462	46054381
32	Kab. Tabanan	60000	362104000	2955330	32649977
33	Kab. Temanggung	5000	30035000	267017	2706165
34	Kab. Trenggalek	4000	25006000	199682	2255137
35	Kab. Tuban	159000	931697000	7666878	84002592
36	Kab. Tulungagung	21000	126457000	1046398	11401071
37	Kodya Denpasar	1911200	11064866000	86304964	998051439
38	Kodya Kediri	15000	89500000	743004	8068880
39	Kodya Madiun	32000	193162000	1537400	17420484
40	Kodya Malang	44000	274383000	2186603	24745210
41	Kodya Mataram	10000	63000000	487596	5682946
42	Kodya Mojokerto	69000	406790000	3241513	36886126
43	Kodya Pasuruan	117000	689123000	5552552	62142399
44	Kodya Probolinggo	1197000	6928871000	53738315	625012251
45	Kodya Surabaya	1730000	10276997000	275472361	909229724
46	Grand Total	8987181	52629820000	618301208	4728318277

Gambar 6.13 Biaya Eksisting Bagian 2

Total biaya eksisting yang akan dibandingkan harus dikurangi PPN dan PBBKB karena total biaya pada metode *stepping stone* tidak disertakan PPN dan PBBKB juga.

Tabel 6.3 Biaya Eksisting

Keterangan	Total	(+)/(-)
Total Keseluruhan	52.629.820.000	(+)
Total PPN	618.301.208	(-)
Total PBBKB	4.728.318.277	(-)
Total Biaya Tanpa PPN dan PBBKB	47.283.200.515	

Dengan demikian dapat dibandingkan biaya eksisting dengan biaya hasil penerapan metode transportasi *least cost & stepping stone*. Seperti pada tabel 6.4

Tabel 6.4 Efisiensi Biaya

Biaya Eksisting	Biaya Metode Transportasi	Efisiensi Biaya
47.283.200.515	42.581.894.560	4.701.305.955

BAB VII KESIMPULAN DAN SARAN

Pada bab ini, dijelaskan mengenai kesimpulan dari semua proses yang telah dilakukan dan saran yang dapat diberikan untuk pengembangan yang lebih baik.

7.1 Kesimpulan

Berdasarkan metode yang digunakan pada penelitian ini, maka dapat diambil kesimpulan bahwa menggunakan metode transportasi *least cost* untuk menyelesaikan masalah pendistribusian HSD PT. Pertamina Patra Niaga Region Jatim Balinus dapat dilakukan dan dengan diteruskan dengan metode *stepping stone* hasil yang didapat akan lebih maksimal. Dalam metode ini, *bunker* ditetapkan menjadi *origin* dan kapasitasnya adalah *supply*, serta permintaan di Kabupaten/Kota yang ada sebagai *destiation* dan jumlah permintaanya adalah *demand*. Setelah dilakukannya metode ini bukan hanya meminimalkan biaya transportasi yang dikeluarkan, tetapi alokasi demand juga tepat dilakukan (semua kebutuhan demand dari masing-masing tujuan terpenuhi). Pembuatan optimasi dengan menggunakan bahasa pemrograman java dan dengan aplikasi Netbeans akan membantu mempermudah proses penghitungan dan proses masukkan data lebih mudah dengan adanya UI yang user friendly. Dengan UI yang lebih fleksibel, diharapkan PT. Pertamina Patra Niaga Region Jatim Balinus dapat menggunakan metode transportasi ini ke dalam proses bisnisnya secara berkala setiap bulan guna meminimalkan biaya distribusi.

7.2 Saran

Saran yang dapat diberikan penulis berdasarkan penelitian yang dilakukan adalah:

1. Penetapan biaya beban distribusi dapat dilakukan lebih baik lagi apabila didapatkan tujuan distribusi secara lengkap bukan hanya dari wilayah kabupaten atau kota saja.

2. Disarankan untuk menambahkan otomatisasi penetapan beban biaya dikarenakan semakin banyaknya tujuan maupun *bunker* yang ada, akan sangat memakan waktu untuk menetapkan beban biaya tersebut.
3. Metode ini bisa lebih dimaksimalkan apabila dilanjutkan dengan penjadwalan agar dapat diketahui kapan demand yang diinginkan sampai pada tujuan yang diinginkan pula.

DAFTAR PUSTAKA

- [1] Badan Pengatur Hilir Minyak dan Gas Bumi. 2016. Konsumsi BBM Nasional Per Tahun [Online]. Available at :. [Accessed 2 Maret 2017]
- [2] Craven, B.D., & Islam S.M.N. 2005. *Operation Research Methods (Related Production, Distribution, and Inventory Management Applications)*. University of Melbourne, Victoria University.
- [3] PT. Pertamina Patra Niaga, 2017. *About Our Company* [Online]. Available at: <http://www.pertaminapatraniaga.com/about-us/history/>. [Accessed 3 Maret 2017]
- [4] Ferguson, S. Thomas. *Linear Programming: A concise Introduction. Introduction, 3-8*.
- [5] Joshi, Rekha Vivek. 2013. *Optimization Technique for Optimization of Three Variables*. Mumbai University, India.
- [6] Kakiay, Thomas J., 2008, *Pemrograman Linier*, Andi Offset, Yogyakarta.
- [7] Saul, I. Gass, & Michael, C. Fu. 2013. *Encyclopedia of Operations Research and Management Science*. Springer US
- [8] Fatimah, Nur Laely. 2015. Implementasi Pengotimalan Biaya Transportasi dengan North West Corner Method (NWCN) dan Stepping Stone Method (SSM) Untuk Distribusi Raskin pada Perum Bulog Sub Divre Semarang. Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Semarang.
- [9] Hariyono, Achmad. 2012. Analisis Penerapan Model Transportasi Distribusi dengan Menggunakan NWCN dan SSM pada Harian Tribun Makassar. Jurusan Manajemen, Fakultas Ekonomi dan Bisnis, Universitas Hasanudin Makassar.
- [10] Miptahudin. 2010. Analisis Perbandingan Pengiriman Barang Menggunakan Metode Transportasi (Studi Kasus di PT. Arta Boga Jakarta). Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif Hidayatullah Jakarta.

- [11] Zhenping Li* , Zhiguo Wu, Lulu Jiang.2016. Research on the Refined Oil Distribution Routing Problem Based on Ant-colony Algorithm. School of Information, Beijing Wuzi University, Beijing, China.
- [12] Mulyono, Sri. 2007. Riset Operasi (Arti Riset Operasi). Lembaga Penerbit FEUI
- [13] Siang, Jok Jek. 2014. Riset Operasi dalam Pendekatan Algoritmis. Penerbit ANDI Yogyakarta.
- [14] Render, Barry & Jay, Hazer. 1997. *Principles of Operation Management*, Second edition, Upper Saddle River, New Jersey, Prentice Hall, Inc.
- [15] Rachman, Taufiqur. 2015. Pengantar Riset Operasional. Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Esa Unggul.
- [16] Taylor III, Bernard W.2013.Introduction to Management Science.Virginia Polytechnic Institute and State University, One Lake Street, Upper Saddle River, New Jersey 07458.

BIODATA PENULIS



Syafriandi Nurrahman, lahir di kota Bontang pada tanggal 20 Agustus 1993, penulis adalah anak ketiga dari tiga bersaudara dari pasangan Moch. Syaiful Anwar dan Wilujeng Harimulyati Agustini. Pendidikan SD penulis dijalani di SD Islam Yabis, Bontang sampai dengan kelas 5. Kelas 6 SD berpindah ke SD Muhammadiyah 4 Pucang, Surabaya. Kemudian SMP Negeri 1 Surabaya dan diteruskan di SMA Negeri 2 Surabaya. Kemudian penulis memilih berkuliah di Institut Teknologi Sepuluh Nopember Jurusan Sistem Informasi masuk pada angkatan 2011. Pada Semester akhir penulis mengambil topik tugas akhir optimasi yang tergabung di lab Rekayasa Data dan Intelegensi Bisnis, Jurusan Sistem Informasi, Fakultas Teknologi Informasi, ITS. E-mail penulis: Syafriandio.oid@gmail.com

Halaman ini sengaja dikosongkan

LAMPIRAN A

DATA REALISASI DESEMBER 2016

1	Sales Organization	Dist Channel	Division	Sales Office	SO Doc Type	No Billing	No Faktur Pajak	SO No	Item SO	Your Reference
15149	3231 (PT PN Marketing BD)	10 (Agent / Distributor)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000761367	1.00342E+14	4018427550	10	8098-DW
15150	3231 (PT PN Marketing BD)	10 (Agent / Distributor)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000762679	1.00342E+14	4018427613	10	8197/SG
15151	3231 (PT PN Marketing BD)	10 (Agent / Distributor)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000761368	1.00342E+14	4018427551	10	8102-DW
15152	3231 (PT PN Marketing BD)	10 (Agent / Distributor)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000762680	1.00342E+14	4018427614	10	8205/SG
15153	3231 (PT PN Marketing BD)	10 (Agent / Distributor)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000759127	1.00342E+14	4018428788	10	8166/SG
15154	3231 (PT PN Marketing BD)	10 (Agent / Distributor)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000762681	1.00342E+14	4018428879	10	8035-DW
15155	3231 (PT PN Marketing BD)	10 (Agent / Distributor)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000759143	1.00342E+14	4018428766	10	8106-HS
15156	3231 (PT PN Marketing BD)	10 (Agent / Distributor)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000762682	1.00342E+14	4018426664	10	8019-DW
15157	3231 (PT PN Marketing BD)	10 (Agent / Distributor)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000759128	1.00342E+14	4018428793	10	8161-HS
15158	3231 (PT PN Marketing BD)	10 (Agent / Distributor)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000762683	1.00342E+14	4018428887	10	8018-HS
15159	3231 (PT PN Marketing BD)	10 (Agent / Distributor)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000762684	1.00342E+14	4018428875	10	8032-DW
15160	3231 (PT PN Marketing BD)	10 (Agent / Distributor)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000762685	1.00342E+14	4018428877	10	8034-DW
15161	3231 (PT PN Marketing BD)	10 (Agent / Distributor)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000762685	1.00342E+14	4018428877	20	8034-DW
15162	3231 (PT PN Marketing BD)	10 (Agent / Distributor)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000764027	1.00342E+14	4018429079	10	8044-HS
15163	3231 (PT PN Marketing BD)	10 (Agent / Distributor)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000759129	1.00342E+14	4018429038	10	8001-DW
15164	3231 (PT PN Marketing BD)	10 (Agent / Distributor)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000764028	1.00342E+14	4018428810	10	8040-HD
15165	3231 (PT PN Marketing BD)	10 (Agent / Distributor)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000759130	1.00342E+14	4018428835	10	8109-HD
15166	3231 (PT PN Marketing BD)	10 (Agent / Distributor)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000759131	1.00342E+14	4018429663	10	8094-HS
15167	3231 (PT PN Marketing BD)	10 (Agent / Distributor)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000759131	1.00342E+14	4018429663	20	8094-HS
15168	3231 (PT PN Marketing BD)	10 (Agent / Distributor)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000769528	1.0342E+14	4018428554	10	8150-HD
15169	3231 (PT PN Marketing BD)	10 (Agent / Distributor)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000762500	1.00342E+14	4018428968	10	8023/SG
15170	3231 (PT PN Marketing BD)	10 (Agent / Distributor)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000764234	1.00342E+14	4018429249	10	8103/SG
15171	3231 (PT PN Marketing BD)	10 (Agent / Distributor)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000764235	1.00342E+14	4018430330	10	8118-HD
15172	3231 (PT PN Marketing BD)	10 (Agent / Distributor)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000764876	1.00342E+14	4018430291	10	8081-HD

Gambar 7.1 Data Realisasi Bag.1

1	Diskon Qty	Diskon	PPH	PBBKB	PPH	Rounding	Total	Currency	Bill Stat	Month	Year	Conc	WILAYAH	SR	Qty (KL)	Total (IDR)	QTY LTR
15149	-9,310,785	0	2,638,511	241,863	0	-479	29,265,000	IDR	Billed	12 Dec	2016	4018427550-10	JATIM BALINUS	SR JATIM 3	5	29,265,000	5,000
15150	-16,160,138	0	4,221,617	376,146	0	69	46,814,000	IDR	Billed	12 Dec	2016	4018427613-10	JATIM BALINUS	SR JATIM 3	8	46,814,000	8,000
15151	-17,350,848	0	4,113,370	365,930	0	4	45,613,000	IDR	Billed	12 Dec	2016	4018427551-10	JATIM BALINUS	SR JATIM 3	8	45,613,000	8,000
15152	-9,987,325	0	2,570,857	236,059	0	-481	28,515,000	IDR	Billed	12 Dec	2016	4018427614-10	JATIM BALINUS	SR JATIM 3	5	28,515,000	5,000
15153	-9,987,325	0	2,570,857	236,059	0	-481	28,515,000	IDR	Billed	12 Dec	2016	4018428788-10	JATIM BALINUS	SR JATIM 3	5	28,515,000	5,000
15154	-17,350,848	0	4,113,370	365,930	0	4	45,613,000	IDR	Billed	12 Dec	2016	4018428879-10	JATIM BALINUS	SR JATIM 3	8	45,613,000	8,000
15155	-477,050	0	2,975,428	274,757	0	-465	33,004,000	IDR	Billed	12 Dec	2016	4018428766-10	JATIM BALINUS	SR JATIM 3	5	33,004,000	5,000
15156	-36,000,656	0	8,226,739	720,714	0	155	91,215,000	IDR	Billed	12 Dec	2016	4018426684-10	JATIM BALINUS	SR JATIM 3	16	91,215,000	16,000
15157	-5,536,300	0	2,638,511	239,929	0	455	29,264,000	IDR	Billed	12 Dec	2016	4018426793-10	JATIM BALINUS	SR JATIM 3	5	29,264,000	5,000
15158	-36,000,656	0	8,226,739	720,714	0	155	91,215,000	IDR	Billed	12 Dec	2016	4018426687-10	JATIM BALINUS	SR JATIM 3	16	91,215,000	16,000
15159	-16,340,544	0	4,113,370	374,598	0	128	45,622,000	IDR	Billed	12 Dec	2016	4018428875-10	JATIM BALINUS	SR JATIM 3	8	45,622,000	8,000
15160	-36,289,312	0	8,226,741	718,238	0	-387	91,212,000	IDR	Billed	12 Dec	2016	4018428877-10	JATIM BALINUS	SR JATIM 3	16	91,212,000	16,000
15161	-36,289,312	0	8,226,741	718,238	0	-387	91,212,000	IDR	Billed	12 Dec	2016	4018428877-20	JATIM BALINUS	SR JATIM 3	16	91,212,000	16,000
15162	-35,567,664	0	8,226,741	724,429	0	421	91,219,000	IDR	Billed	12 Dec	2016	4018429579-10	JATIM BALINUS	SR JATIM 3	16	91,219,000	16,000
15163	-36,289,312	0	8,226,741	718,238	0	-387	91,212,000	IDR	Billed	12 Dec	2016	4018429698-10	JATIM BALINUS	SR JATIM 3	16	91,212,000	16,000
15164	-9,085,270	0	2,638,511	243,798	0	-414	29,267,000	IDR	Billed	12 Dec	2016	4018428810-10	JATIM BALINUS	SR JATIM 3	5	29,267,000	5,000
15165	-9,987,325	0	2,570,857	236,059	0	-481	28,515,000	IDR	Billed	12 Dec	2016	4018428935-10	JATIM BALINUS	SR JATIM 3	5	28,515,000	5,000
15166	-36,289,312	0	8,226,741	718,238	0	-387	91,212,000	IDR	Billed	12 Dec	2016	4018429663-10	JATIM BALINUS	SR JATIM 3	16	91,212,000	16,000
15167	-36,289,312	0	8,226,741	718,238	0	-387	91,212,000	IDR	Billed	12 Dec	2016	4018429663-20	JATIM BALINUS	SR JATIM 3	16	91,212,000	16,000
15168	-1,018,176	0	8,676,163	0	0	5	95,440,000	IDR	Billed	12 Dec	2016	4018428854-10	JATIM BALINUS	SR JATIM 3	16	95,440,000	16,000
15169	-9,085,270	0	2,683,614	243,798	0	453	29,764,000	IDR	Billed	12 Dec	2016	4018428968-10	JATIM BALINUS	SR JATIM 3	5	29,764,000	5,000
15170	-9,761,830	0	2,555,671	237,994	0	230	28,344,000	IDR	Billed	12 Dec	2016	4018429249-10	JATIM BALINUS	SR JATIM 3	5	28,344,000	5,000
15171	-15,799,312	0	4,196,159	379,242	0	-193	46,539,000	IDR	Billed	12 Dec	2016	4018430330-10	JATIM BALINUS	SR JATIM 3	8	46,539,000	8,000
15172	-9,468,645	0	2,632,725	240,509	0	-479	29,090,000	IDR	Billed	12 Dec	2016	4018430291-10	JATIM BALINUS	SR JATIM 3	5	29,090,000	5,000

Gambar 7.2 Data Realisasi Bag.1 ke 2

1	Sales Organization	Dist Channel	Division	Sales Office	SO Doc Type	No Billing	No Faktur Pajak	SO No	Item SO	Your Reference
15195	3231 PT PN Marketing BD	10 Agent / Distributor	04 Fuel	3101 GM III JATENG DIY JA	ZTS (Third Party Order)	3000767164	1.00342E+14	4018432449	10	D024-DW
15196	3231 PT PN Marketing BD	10 Agent / Distributor	04 Fuel	3101 GM III JATENG DIY JA	ZTS (Third Party Order)	3000767164	1.00342E+14	4018432449	20	D024-DW
15197	3231 PT PN Marketing BD	10 Agent / Distributor	04 Fuel	3101 GM III JATENG DIY JA	ZTS (Third Party Order)	3000767164	1.00342E+14	4018432449	30	D024-DW
15198	3231 PT PN Marketing BD	10 Agent / Distributor	04 Fuel	3101 GM III JATENG DIY JA	ZTS (Third Party Order)	3000767164	1.00342E+14	4018432449	40	D024-DW
15199	3231 PT PN Marketing BD	10 Agent / Distributor	04 Fuel	3101 GM III JATENG DIY JA	ZTS (Third Party Order)	3000767968	1.00342E+14	4018432150	10	D047-HD
15200	3231 PT PN Marketing BD	10 Agent / Distributor	04 Fuel	3101 GM III JATENG DIY JA	ZTS (Third Party Order)	3000767968	1.00342E+14	4018432459	10	D025-DW
15201	3231 PT PN Marketing BD	10 Agent / Distributor	04 Fuel	3101 GM III JATENG DIY JA	ZTS (Third Party Order)	3000767968	1.00342E+14	4018433062	10	D023-NN
15202	3231 PT PN Marketing BD	10 Agent / Distributor	04 Fuel	3101 GM III JATENG DIY JA	ZTS (Third Party Order)	3000767967	1.00342E+14	4018432708	10	D028-HD
15203	3231 PT PN Marketing BD	10 Agent / Distributor	04 Fuel	3101 GM III JATENG DIY JA	ZTS (Third Party Order)	3000766186	1.00342E+14	4018431997	10	D012-DW
15204	3231 PT PN Marketing BD	10 Agent / Distributor	04 Fuel	3101 GM III JATENG DIY JA	ZTS (Third Party Order)	3000766186	1.00342E+14	4018431997	20	D012-DW
15205	3231 PT PN Marketing BD	10 Agent / Distributor	04 Fuel	3101 GM III JATENG DIY JA	ZTS (Third Party Order)	3000768060	1.00342E+14	4018432137	10	D046-HD
15206	3231 PT PN Marketing BD	10 Agent / Distributor	04 Fuel	3101 GM III JATENG DIY JA	ZTS (Third Party Order)	3000767968	1.00342E+14	4018432152	10	D049-HD
15207	3231 PT PN Marketing BD	10 Agent / Distributor	04 Fuel	3101 GM III JATENG DIY JA	ZTS (Third Party Order)	3000767969	1.00342E+14	4018432426	10	D022-DW
15208	3231 PT PN Marketing BD	10 Agent / Distributor	04 Fuel	3101 GM III JATENG DIY JA	ZTS (Third Party Order)	3000767970	1.00342E+14	4018432433	10	D016-DW
15209	3231 PT PN Marketing BD	10 Agent / Distributor	04 Fuel	3101 GM III JATENG DIY JA	ZTS (Third Party Order)	3000767163	1.00342E+14	4018432155	10	D040-HD
15210	3231 PT PN Marketing BD	10 Agent / Distributor	04 Fuel	3101 GM III JATENG DIY JA	ZTS (Third Party Order)	3000767166	1.00342E+14	4018432161	10	D029-HD
15211	3231 PT PN Marketing BD	10 Agent / Distributor	04 Fuel	3101 GM III JATENG DIY JA	ZTS (Third Party Order)	3000767971	1.00342E+14	4018432466	10	D029-DW
15212	3231 PT PN Marketing BD	10 Agent / Distributor	04 Fuel	3101 GM III JATENG DIY JA	ZTS (Third Party Order)	3000766187	1.00342E+14	4018431977	10	D008-DW
15213	3231 PT PN Marketing BD	10 Agent / Distributor	04 Fuel	3101 GM III JATENG DIY JA	ZTS (Third Party Order)	3000767972	1.00342E+14	4018432157	10	D015-HD
15214	3231 PT PN Marketing BD	10 Agent / Distributor	04 Fuel	3101 GM III JATENG DIY JA	ZTS (Third Party Order)	3000768061	1.00342E+14	4018431621	10	D002-HD
15215	3231 PT PN Marketing BD	10 Agent / Distributor	04 Fuel	3101 GM III JATENG DIY JA	ZTS (Third Party Order)	3000768062	1.00342E+14	4018431618	10	D001-HD
15216	3231 PT PN Marketing BD	10 Agent / Distributor	04 Fuel	3101 GM III JATENG DIY JA	ZTS (Third Party Order)	3000768041	1.00342E+14	4018431956	10	D005-DW
15217	3231 PT PN Marketing BD	10 Agent / Distributor	04 Fuel	3101 GM III JATENG DIY JA	ZTS (Third Party Order)	3000768042	1.00342E+14	4018431633	10	D006-HD
15218	3231 PT PN Marketing BD	10 Agent / Distributor	04 Fuel	3101 GM III JATENG DIY JA	ZTS (Third Party Order)	3000768043	1.00342E+14	4018431654	10	D017-HD

Gambar 7.3 Data Realisasi Bag.2

Diskon	PPn	PBEK3	PPn	Rounding	Total	Currency	Bill Stat	Month	Year	Conc	WILAYAH	SR	Qty (KL)	Total (IDR)	QTY LTR
0	4,088,112	360,357	0	411	45,330,000	IDR	Billed	12 Dec	2016	4018432449-10	JATIM BALINUS	SR JATIM 3	8	45,330,000	8,000
0	4,088,112	360,357	0	411	45,330,000	IDR	Billed	12 Dec	2016	4018432449-20	JATIM BALINUS	SR JATIM 3	8	45,330,000	8,000
0	4,088,112	360,357	0	411	45,330,000	IDR	Billed	12 Dec	2016	4018432449-30	JATIM BALINUS	SR JATIM 3	8	45,330,000	8,000
0	4,088,112	360,357	0	411	45,330,000	IDR	Billed	12 Dec	2016	4018432449-40	JATIM BALINUS	SR JATIM 3	8	45,330,000	8,000
0	4,088,113	359,119	0	-360	45,328,000	IDR	Billed	12 Dec	2016	4018432150-10	JATIM BALINUS	SR JATIM 3	8	45,328,000	8,000
0	2,622,725	239,929	0	101	29,090,000	IDR	Billed	12 Dec	2016	4018432459-10	JATIM BALINUS	SR JATIM 3	5	29,090,000	5,000
0	4,196,359	376,146	0	-97	46,536,000	IDR	Billed	12 Dec	2016	4018433082-10	JATIM BALINUS	SR JATIM 3	8	46,536,000	8,000
0	8,176,226	718,238	0	280	90,657,000	IDR	Billed	12 Dec	2016	4018432708-10	JATIM BALINUS	SR JATIM 3	16	90,657,000	16,000
0	8,176,226	718,238	0	280	90,657,000	IDR	Billed	12 Dec	2016	4018431997-10	JATIM BALINUS	SR JATIM 3	16	90,657,000	16,000
0	8,176,226	718,238	0	280	90,657,000	IDR	Billed	12 Dec	2016	4018431997-20	JATIM BALINUS	SR JATIM 3	16	90,657,000	16,000
0	2,683,614	243,798	0	453	29,764,000	IDR	Billed	12 Dec	2016	4018432137-10	JATIM BALINUS	SR JATIM 3	5	29,764,000	5,000
0	2,622,725	236,059	0	-29	29,086,000	IDR	Billed	12 Dec	2016	4018432152-10	JATIM BALINUS	SR JATIM 3	5	29,086,000	5,000
0	2,555,071	236,059	0	165	28,342,000	IDR	Billed	12 Dec	2016	4018432426-10	JATIM BALINUS	SR JATIM 3	5	28,342,000	5,000
0	4,088,113	365,311	0	448	45,335,000	IDR	Billed	12 Dec	2016	4018432433-10	JATIM BALINUS	SR JATIM 3	8	45,335,000	8,000
0	8,176,224	720,714	0	-178	90,659,000	IDR	Billed	12 Dec	2016	4018432155-10	JATIM BALINUS	SR JATIM 3	16	90,659,000	16,000
0	8,176,226	718,238	0	280	90,657,000	IDR	Billed	12 Dec	2016	4018432161-10	JATIM BALINUS	SR JATIM 3	16	90,657,000	16,000
0	4,088,113	365,311	0	448	45,335,000	IDR	Billed	12 Dec	2016	4018432466-10	JATIM BALINUS	SR JATIM 3	8	45,335,000	8,000
0	4,088,113	363,143	0	-384	45,332,000	IDR	Billed	12 Dec	2016	4018431977-10	JATIM BALINUS	SR JATIM 3	8	45,332,000	8,000
0	2,555,071	232,189	0	35	28,338,000	IDR	Billed	12 Dec	2016	4018432157-10	JATIM BALINUS	SR JATIM 3	5	28,338,000	5,000
0	2,683,614	243,798	0	453	29,764,000	IDR	Billed	12 Dec	2016	4018431621-10	JATIM BALINUS	SR JATIM 3	5	29,764,000	5,000
0	2,683,614	236,059	0	192	29,756,000	IDR	Billed	12 Dec	2016	4018431618-10	JATIM BALINUS	SR JATIM 3	5	29,756,000	5,000
0	4,088,113	363,143	0	-384	45,332,000	IDR	Billed	12 Dec	2016	4018431956-10	JATIM BALINUS	SR JATIM 3	8	45,332,000	8,000
0	8,176,226	713,903	0	-385	90,652,000	IDR	Billed	12 Dec	2016	4018431633-10	JATIM BALINUS	SR JATIM 3	16	90,652,000	16,000
0	2,555,071	232,189	0	35	28,338,000	IDR	Billed	12 Dec	2016	4018431654-10	JATIM BALINUS	SR JATIM 3	5	28,338,000	5,000

Gambar 7.4 Data Realisasi Bag.2 ke 2

1	Sales Organization	Dist Channel	Division	Sales Office	SO Doc Type	No Billing	No Faktur Pajak	SO No	Item SO	Your Reference
15782	3231 (PT PN Marketing BD)	20 (Direct)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000761933	1.00342E+14	4018428677	100	087-AS
15783	3231 (PT PN Marketing BD)	20 (Direct)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000761933	1.00342E+14	4018428677	110	087-AS
15784	3231 (PT PN Marketing BD)	20 (Direct)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000761933	1.00342E+14	4018428677	120	087-AS
15785	3231 (PT PN Marketing BD)	20 (Direct)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000761933	1.00342E+14	4018428677	130	087-AS
15786	3231 (PT PN Marketing BD)	20 (Direct)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000762717	1.00342E+14	4018428677	140	087-AS
15787	3231 (PT PN Marketing BD)	20 (Direct)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000762717	1.00342E+14	4018428677	150	087-AS
15788	3231 (PT PN Marketing BD)	20 (Direct)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000762717	1.00342E+14	4018428677	160	087-AS
15789	3231 (PT PN Marketing BD)	20 (Direct)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000762717	1.00342E+14	4018428677	170	087-AS
15790	3231 (PT PN Marketing BD)	20 (Direct)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000762717	1.00342E+14	4018428677	180	087-AS
15791	3231 (PT PN Marketing BD)	20 (Direct)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000762717	1.00342E+14	4018428677	190	087-AS
15792	3231 (PT PN Marketing BD)	20 (Direct)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000761934	1.00342E+14	4018428677	200	087-AS
15793	3231 (PT PN Marketing BD)	20 (Direct)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000769258	1.00342E+14	4018434351	10	D098-DW
15794	3231 (PT PN Marketing BD)	20 (Direct)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000769258	1.00342E+14	4018434351	20	D098-DW
15795	3231 (PT PN Marketing BD)	20 (Direct)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000769258	1.00342E+14	4018434351	30	D098-DW
15796	3231 (PT PN Marketing BD)	20 (Direct)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000769258	1.00342E+14	4018434351	40	D098-DW
15797	3231 (PT PN Marketing BD)	20 (Direct)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000768676	1.00342E+14	4018434351	50	D098-DW
15798	3231 (PT PN Marketing BD)	20 (Direct)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000768676	1.00342E+14	4018434351	60	D098-DW
15799	3231 (PT PN Marketing BD)	20 (Direct)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000773045	1.00342E+14	4018438787	10	D068-HD
15800	3231 (PT PN Marketing BD)	20 (Direct)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000773045	1.00342E+14	4018438787	20	D068-HD
15801	3231 (PT PN Marketing BD)	20 (Direct)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000772726	1.00342E+14	4018438787	30	D068-HD
15802	3231 (PT PN Marketing BD)	20 (Direct)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000772726	1.00342E+14	4018438787	40	D068-HD
15803	3231 (PT PN Marketing BD)	20 (Direct)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000772726	1.00342E+14	4018438787	50	D068-HD
15804	3231 (PT PN Marketing BD)	20 (Direct)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000772726	1.00342E+14	4018438787	60	D068-HD
15805	3231 (PT PN Marketing BD)	20 (Direct)	04 (Fuel)	3101 (GM III JATENG DIY JA)	ZTS (Third Party Order)	3000772726	1.00342E+14	4018438787	70	D068-HD

Gambar 7.5 Data Realisasi Bag.3

Diskon Qty	Diskon	PPH	PBB	PPH	Rounding	Total	Currency	Bill Stat	Month	Year	Conc	WILAYAH	SR	Qty (KL)	Total (IDR)	QTY LTR
-11,791,435	0	2,570,857	220,579	0	-1	28,500,000	IDR	Billed	12 Dec	2016	4018428677-100	JATIM BALINUS	SR BALINUS	5	28,500,000	5,000
-11,791,435	0	2,570,857	220,579	0	-1	28,500,000	IDR	Billed	12 Dec	2016	4018428677-110	JATIM BALINUS	SR BALINUS	5	28,500,000	5,000
-11,791,435	0	2,570,857	220,579	0	-1	28,500,000	IDR	Billed	12 Dec	2016	4018428677-120	JATIM BALINUS	SR BALINUS	5	28,500,000	5,000
-11,791,435	0	2,570,857	220,579	0	-1	28,500,000	IDR	Billed	12 Dec	2016	4018428677-130	JATIM BALINUS	SR BALINUS	5	28,500,000	5,000
-11,791,435	0	2,570,857	220,579	0	-1	28,500,000	IDR	Billed	12 Dec	2016	4018428677-140	JATIM BALINUS	SR BALINUS	5	28,500,000	5,000
-11,791,435	0	2,570,857	220,579	0	-1	28,500,000	IDR	Billed	12 Dec	2016	4018428677-150	JATIM BALINUS	SR BALINUS	5	28,500,000	5,000
-11,791,435	0	2,570,857	220,579	0	-1	28,500,000	IDR	Billed	12 Dec	2016	4018428677-160	JATIM BALINUS	SR BALINUS	5	28,500,000	5,000
-11,791,435	0	2,570,857	220,579	0	-1	28,500,000	IDR	Billed	12 Dec	2016	4018428677-170	JATIM BALINUS	SR BALINUS	5	28,500,000	5,000
-11,791,435	0	2,570,857	220,579	0	-1	28,500,000	IDR	Billed	12 Dec	2016	4018428677-180	JATIM BALINUS	SR BALINUS	5	28,500,000	5,000
-11,791,435	0	2,570,857	220,579	0	-1	28,500,000	IDR	Billed	12 Dec	2016	4018428677-190	JATIM BALINUS	SR BALINUS	5	28,500,000	5,000
-11,791,435	0	2,570,857	220,579	0	-1	28,500,000	IDR	Billed	12 Dec	2016	4018428677-200	JATIM BALINUS	SR BALINUS	5	28,500,000	5,000
-11,889,380	0	2,661,062	228,319	0	-1	29,500,000	IDR	Billed	12 Dec	2016	4018434351-10	JATIM BALINUS	SR BALINUS	5	29,500,000	5,000
-11,889,380	0	2,661,062	228,319	0	-1	29,500,000	IDR	Billed	12 Dec	2016	4018434351-20	JATIM BALINUS	SR BALINUS	5	29,500,000	5,000
-11,889,380	0	2,661,062	228,319	0	-1	29,500,000	IDR	Billed	12 Dec	2016	4018434351-30	JATIM BALINUS	SR BALINUS	5	29,500,000	5,000
-11,889,380	0	2,661,062	228,319	0	-1	29,500,000	IDR	Billed	12 Dec	2016	4018434351-40	JATIM BALINUS	SR BALINUS	5	29,500,000	5,000
-11,889,380	0	2,661,062	228,319	0	-1	29,500,000	IDR	Billed	12 Dec	2016	4018434351-50	JATIM BALINUS	SR BALINUS	5	29,500,000	5,000
-11,889,380	0	2,661,062	228,319	0	-1	29,500,000	IDR	Billed	12 Dec	2016	4018434351-60	JATIM BALINUS	SR BALINUS	5	29,500,000	5,000
-11,889,380	0	2,661,062	228,319	0	-1	29,500,000	IDR	Billed	12 Dec	2016	4018438787-10	JATIM BALINUS	SR BALINUS	5	29,500,000	5,000
-11,889,380	0	2,661,062	228,319	0	-1	29,500,000	IDR	Billed	12 Dec	2016	4018438787-20	JATIM BALINUS	SR BALINUS	5	29,500,000	5,000
-11,889,380	0	2,661,062	228,319	0	-1	29,500,000	IDR	Billed	12 Dec	2016	4018438787-30	JATIM BALINUS	SR BALINUS	5	29,500,000	5,000
-11,889,380	0	2,661,062	228,319	0	-1	29,500,000	IDR	Billed	12 Dec	2016	4018438787-40	JATIM BALINUS	SR BALINUS	5	29,500,000	5,000
-11,889,380	0	2,661,062	228,319	0	-1	29,500,000	IDR	Billed	12 Dec	2016	4018438787-50	JATIM BALINUS	SR BALINUS	5	29,500,000	5,000
-11,889,380	0	2,661,062	228,319	0	-1	29,500,000	IDR	Billed	12 Dec	2016	4018438787-60	JATIM BALINUS	SR BALINUS	5	29,500,000	5,000
-11,889,380	0	2,661,062	228,319	0	-1	29,500,000	IDR	Billed	12 Dec	2016	4018438787-70	JATIM BALINUS	SR BALINUS	5	29,500,000	5,000

Gambar 7.6 Data Realisasi Bag.3 ke 2

LAMPIRAN B

PRAPROSES TABEL PIVOT

Row Labels	Sum of QTY LTR	Sum of Total (IDR)	Sum of PBBKB	Sum of PPN
Kab. Badung	146180	876141000	7415711	78975135
PT. AIRLANGGA PUTRA MANDIRI	65000	397810000	3231294	35870795
12/1/2016	10000	60000000	487596	5410220
12/5/2016	10000	60000000	487596	5410220
12/6/2016	10000	60000000	487596	5410220
12/7/2016	10000	60000000	487596	5410220
12/28/2016	15000	94686000	768546	8537949
12/29/2016	10000	63124000	512364	5691966
PT. PUTRA SANI IWANI	72680	425393000	3642642	38341075
12/2/2016	5000	28500000	263147	2566987
12/3/2016	2000	11400000	108355	1026513
12/5/2016	9000	51300000	471343	4620788
12/6/2016	10000	57000000	506946	5135733
12/15/2016	5000	29736000	290236	2676848
12/19/2016	7000	41584000	360667	3747587
12/21/2016	1000	5937000	47985	535370
12/22/2016	8000	47494000	381564	4282957
12/24/2016	5000	29681000	236059	2676848
12/26/2016	680	4037000	32104	364051
12/27/2016	1000	5936000	47212	535370
12/28/2016	15000	89043000	708177	8030544

Gambar 8.1 Data Tabel Pivot Bag.1

⇒ Kab. Bangkalan	82000	485267000	4015318	43750115
⇒ PT. ENERGI NIAGA UTAMA	5000	27809000	223675	2507713
12/10/2016	5000	27809000	223675	2507713
⇒ PT. TRIJAYA ADYMIK	77000	457458000	3791643	41242402
12/2/2016	8000	46947000	390077	4232442
12/5/2016	8000	46947000	390077	4232442
12/6/2016	5000	29342000	243798	2645276
12/7/2016	8000	46947000	390077	4232442
12/9/2016	8000	46947000	390077	4232442
12/12/2016	8000	46947000	390077	4232442
12/17/2016	8000	48345000	399365	4358729
12/19/2016	8000	48345000	399365	4358729
12/22/2016	8000	48345000	399365	4358729
12/25/2016	5000	30216000	249603	2724206
12/28/2016	3000	18130000	149762	1634523
⇒ Kab. Banyuwangi	183000	1096225000	15247971	98270691
⇒ PT. BAHARI NUSANTARA	7000	43001000	1869609	3739217
12/5/2016	7000	43001000	1869609	3739217
⇒ PT. CITRA SEMESTA ENERGY	10000	56696000	483726	5110142
12/6/2016	5000	28348000	241863	2555071
12/13/2016	5000	28348000	241863	2555071
⇒ PT. ENERGI NIAGA UTAMA	20000	117632000	942300	10608165
12/17/2016	15000	88224000	707015	7956123
12/28/2016	5000	29408000	235285	2652042
⇒ PT. INDOLINE INCOMEKITA	23000	135850000	5906523	11813044

Gambar 8.2 Data Tabel Pivot Bag.2

Material Description	PATRA DIESEL/HSD - PTM			
Row Labels	Sum of QTY LTR	Sum of Total (IDR)	Sum of PBBKB	Sum of PPN
* Kab. Badung	146180	876141000	7415711	78975135
* Kab. Bangkalan	82000	485267000	4015318	43750115
* Kab. Banyuwangi	183000	1096225000	15247971	98270691
* Kab. Bojonegoro	5000	27816000	231415	2507713
* Kab. Gianyar	4850	28320000	242018	2552598
* Kab. Gresik	1196000	6852910000	54528925	618032745
* Kab. Jember	43000	278001000	2223596	25070814
* Kab. Jembrana	45000	265070000	2314147	23887011
* Kab. Jombang	82000	486127000	3933666	43835810
* Kab. Kediri	15000	89774000	743004	8093688
* Kab. Lamongan	23000	137943000	1135403	12437082
* Kab. Lumajang	5000	28340000	234124	2555071
* Kab. Madiun	117000	732575000	5927401	66058378
* Kab. Malang	26000	147342000	1195620	13285916
* Kab. Mojokerto	664951	3774827000	29383218	340494822
* Kab. Nganjuk	5000	31008000	247668	2796370
* Kab. Ngawi	29000	172125000	1411940	15519403
* Kab. Pacitan	10000	65516000	522425	5908460
* Kab. Pamekasan	5000	30962000	1429348	2684783
* Kab. Pasuruan	121000	710875000	5768114	64100473
* Kab. Ponorogo	5000	31008000	247668	2796370

Gambar 8.3 Data Tabel Pivot Bag.3

⊕ Kab. Ponorogo	5000	31008000	247668	2796370
⊕ Kab. Probolinggo	44000	259979000	2150843	23438995
⊕ Kab. Rembang	8000	49098000	398127	4427286
⊕ Kab. Sidoarjo	534000	3234468000	26003711	291678358
⊕ Kab. Situbondo	29000	161315000	1322084	14544733
⊕ Kab. Sumbawa	100000	604250000	4676668	54506675
⊕ Kab. Sumbawa Barat	85000	510547000	3951462	46054381
⊕ Kab. Tabanan	60000	362104000	2955330	32649977
⊕ Kab. Temanggung	5000	30035000	267017	2706165
⊕ Kab. Trenggalek	4000	25006000	199682	2255137
⊕ Kab. Tuban	159000	931697000	7666878	84002592
⊕ Kab. Tulungagung	21000	126457000	1046398	11401071
⊕ Kodya Denpasar	1911200	11064866000	86304964	998051439
⊕ Kodya Kediri	15000	89500000	743004	8068880
⊕ Kodya Madiun	32000	193162000	1537400	17420484
⊕ Kodya Malang	44000	274383000	2186603	24745210
⊕ Kodya Mataram	10000	63000000	487596	5682946
⊕ Kodya Mojokerto	69000	406790000	3241513	36686126
⊕ Kodya Pasuruan	117000	689123000	5552552	62142399
⊕ Kodya Probolinggo	1197000	6928871000	53738315	625012251
⊕ Kodya Surabaya	1730000	10276997000	275472361	909229724
Grand Total	8987181	52629820000	618301208	4728318277

Gambar 8.4 Data Tabel Pivot Bag.4